A QUANTITATIVE AND MATHEMATICAL ANALYSIS OF CELL MOTILITY: ADHESION-MEDIATED THREE-DIMENSIONAL TISSUE CELL MIGRATION AND *Listeria monocytogenes*

By

BRIAN TOY BURGESS

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2000

This work and contribution to science would not exist without the support and love from my family. This work is dedicated to my parents, Toy Gerald and Dianne, my brother and sister, Kevin and Ali, and to my lovely wife, Melanie Flynne Burgess. In addition to God, their joy, support and love has been an unending source of strength and encouragement.

ACKNOWLEDGMENTS

It has been a wonderful experience to work with so many talented persons. I feel privileged to have been afforded this unique opportunity. Among those persons, I would like to give special thanks to my co-mentors, Dr. Richard Dickinson and Dr. Daniel Purich. Rich has not only taught me much in regards to learning how to critically evaluate and scientifically investigate, but has also become a trusted friend. Dr. Purich not only taught me much about biophysical world, but the enthusiasm he brings to scientific research continues to impress me and consequently has made our collaboration truly an enjoyment. I would also like to thank the Dr. Roger Tran-Son-Tay and Dr. Spyros Svoronos who both have not only served as wonderful teachers to me throughout my graduate education, but I also have enjoyed working and getting to know them both. Although not on my doctoral committee, I would like to thank Dr. Fred Southwick for his tremendous help and support with the microinjection experiments and for graciously opening his laboratory facilities to complete the microinjection experiments. I would also like to specially thank Dr. Jennifer Myles whom I worked extensively with on the first part of my thesis. Much of this work would not have been completed without her diligent efforts. I would like to especially thank Wei Li for her help in cell culturing, general advice in the lab, and the microinjection experiments. I would like to thank Bruce Gibson for all of his dedicated work involved in the purification of alpha-actinin used in the microinjection experiments. Both Wei and Bruce were an integral part to the success of the Listeria microinjection experiments. Finally, I would like to thank the sweetheart of my life, Melanie, who brings a great friendship, constant kindness, love and support into my life.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

ABSTRACT vii

CHAPTERS

1 INTRODUCTION 1

Cell Motiltiy 1
Tissue Cell Adhesion 3
The Cellular Cytoskeleton 3
Actin and Actin-Binding Proteins 4
Goals and Objective 6

2 QUANTITATIVE ANALYSIS OF ADHESION MEDIATED CELL MIGRATION IN THREE-DIMENSIONAL GELS OF RGD-GRAFTED COLLAGEN 9

Introduction 9
Materials and Methods 13
Results 18
Discussion 23

3 ADHESIVE GRADIENT: A POTENTIAL DIRECTIONAL CUE FOR CELL MIGRATION 29

Introduction 29
Materials and Methods 33
Results 42
Discussion 49

4 THE "ACTIN FILAMENT ZIPPER": A BIOPHYSICAL MODEL FOR ACTIN-BASED MECHANOCHEMICAL PROPULSION 54

Introduction 54
Mechanochemical Model for the Actin-Zipper Mechanism 59
Actin Bundling Reaction Kinetics 60
Actin Bundling and Competitive Binding 64
Formulation of Thermodynamic AFZ Model 66
Equilibrium Model Force Predictions 69
Formulation of Brownian Dynamics AFZ Model 72
Brownian Dynamic Model Force Predictions 77
Discussion 80

5 A QUANTITATIVE STUDY RELATING THE INTRACELLULAR PROPULSION OF *Listeria monocytogenes* AND ALPHA-ACTININ CONCENTRATION 84

Introduction 84
Materials and Methods 86
Results 89
Discussion 92

6 SUMMARY AND FUTURE DIRECTIONS 97

Summary 97
Future Directions 101

APPENDICES

A THREE-DIMENSIONAL CELL TRACKING PROGRAMS 105

Bscan3.mac 105
Bcomm1.mac 110
Bscinic.mac 118

B CELL TRACKING ANALYSIS PROGRAMS 128

D2cold.m 128
Fitexp.m 132
Eta.m 135
Omega.m 136
Phim.m 138
Velem.m 139
Vmatp.m 141

C GRADIENT CELL TRACKING ANALYSIS PROGRAMS 143

Tracksc.m 143
Klinotxs.m 154
Orthotxs.m 157
Fuldxdy.m 160
Turnan.m 165

D COMPUTER CODE FOR THERMODYNAMIC ACTIN FILAMENT ZIPPER MODEL 168

Masinh.m 168
Inhbbend.m 169
Inhbend.m 171
Inhfbend.m 172
Funcv.m 172

E COMPUTER CODE FOR BROWNIAN DYNAMICS ACTIN FILAMENT ZIPPER MODEL 181

Bsimple12.m 181
Chem.m 195
Xlink.m 197

F COMPUTER CODE FOR ANALYZING LISTERIA MOTILITY 200
Listpla2.mac 200
Anzelis2.mac 201
G DERIVATION OF LOCAL BENDING ENERGY, F_{BJ} 205
H BUNDLING VELOCITY OF ALPHA-ACTININ 207
REFERENCES 208
BIOGRAPHICAL SKETCH 218

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A QUANTITATIVE AND MATHEMATICAL ANALYSIS OF CELL MOTILITY: ADHESION-MEDIATED THREE-DIMENSIONAL TISSUE CELL MIGRATION AND *Listeria monocytogenes*

By

Brian Toy Burgess

December 2000

Chair: Richard B. Dickinson
Major Department: Chemical Engineering Department

Cellular locomotion is an integral part of a number of bioprocesses such as wound healing, embryogenesis and the spreading of infectious diseases. A fundamental understanding of how cells are able to move can lead to improved design of a number of tissue-engineering devices and, in the case of the spreading of infectious diseases, specific therapeutic targeting of critically important cellular components involved in the cell-to-cell spreading. In this thesis we investigate two important areas in cell motility: a) adhesion-mediated three-dimensional tissue cell migration and b) the interaction of actin filaments and the actin-bundling protein α-actinin.

In the first study, using a novel three-dimensional collagen-based cell migration assay, we investigated the relationship between the adhesiveness of the cell-matrix and cell motility. By directly observing and recording the time-incremented movements of cells, we found the adhesiveness of the substratum can play a significant and important role in determining how effective the cell is able to migrate. Secondly, by constructing a three-dimensional adhesion gradient, we found one could potentially exploit the relationship between adhesion and migration to direct cellular displacements. These studies are particularly important because they represent

the first studies to directly observe and statistically relate three-dimensional cell migration as a function of increasing adhesiveness and adhesion gradient steepness.

The second study focused on the interaction between actin filaments and actinin in the context of how this interaction relates potentially to the propulsion mechanism utilized by Listeria. We investigated this interaction by formulating both a thermodynamic and Brownian dynamic mathematical model. Our models accounted for the binding kinetics between actin filaments and α-actinin, chemical bond stiffness and the mechanical properties of actin filaments. A key result from these models was that cross-bridging of actin filaments via actinin can dynamically align actin filaments to collinear bundles as similarly observed *in vivo* and that through this bundling mechanism, significant cellular forces can be generated. Lastly, we quantitatively measured the dose-response of Listeria motility to increasing concentrations of actinin. Our results showed a statistically significant biphasic dependence of Listeria motility to increasing concentrations of α-actinin. The results underscore the importance of α-actinin in the intracellular propulsion of Listeria and qualitatively agree with our theoretical model predictions.

CHAPTER 1
INTRODUCTION

Cell Motility

Cell motility is outwardly manifested in a number of ways such as the adhesion-mediated crawling of tissue cells, propulsion via extracellular appendages such as flagellum in sperm cell motility, and through the formation of an actin rocket-like tail in the case of *Listeria monocytogenes*. Functionally, cell motility serves as a fundamental process central to a number of physiological and pathological processes such as embryogenesis, angiogenesis, wound healing, and the spreading of infectious diseases. A greater understanding of how cells are able to produce cellular movements, be it prokaryote (e.g., bacteria) or eukaryotic (e.g., animal cells) cells, has potentially immense benefits both to the scientific and medical communities. In this thesis we will focus on two key aspects of cell motility: 1) adhesion-mediated tissue cell motility and 2) the intracellular actin-based propulsion of *Listeria monocytogenes*.

Tissue cell motility exhibited by such cells as fibroblast cells (give rise to connective tissue) and melanoma cells (skin tumor cells), can be described as a three-step process: 1) attachment to the cell to the substratum, 2) force-generation to pull the cell across the substratum and 3) rupture of existing chemical bond attachments at the rear of the cell. Cell adhesion to the substratum functions in part to provide both anchorage and traction to the cell. Adhesion is mediated by the specific binding between cell surface receptors and complementary adhesion ligands attached to the substratum. The aim of this part of the work is to quantitatively

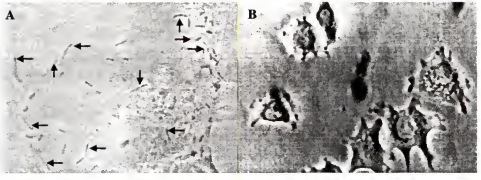

Fig. 1.1. A) Intracellular propulsion of *Listeria monocytogenes* within the cytoplasm of PTK2 cells. Arrows note phase dense actin rocket tails of Listeria. Image at 63x magnification. B) Integrin-mediated fibroblast adhesion to fibronectin coated surface. Image at 300x magnification. Image taken from (Lin et al., 1997).

relate the adhesive properties of the cellular matrix to the three-dimensional tissue cell motility.

In the second part of our work, we study the actin-based intracellular motility of *Listeria monocytogenes*. Listeria is a ubiquitous rod-shaped gram positive pathogenic bacterium that can cause food-borne infections such as listeriosis, with symptoms including septicemia, meningoencephalitis, encephalitis, and intrauterine or cervical infections in pregnant women, which may result in spontaneous abortion (2nd/3rd trimester) or stillbirth. Food-borne listeriosis outbreaks have an average case fatality rate of thirty-percent (Federal Drug Agency, 1999). Once ingested, *Listeria monocytogenes* can be phagocytosed by epithelium cells in the intestinal tract where the bacterium both grow directly in the cytoplasm of the host, releasing virulent factors. To facilitate cell-to-cell spreading, the bacterium recruits, assembles and polymerizes host cytoskeleton proteins forming an actin rocket-like tail at one pole of the bacterium for the purpose of continuously propelling the bacterium through the cell membrane forming long filopodia-like protrusions that can be engulfed by neighboring cells (Southwick and Purich, 1996; Southwick and Purich, 1994; Dabiri et al., 1990). The aim of this research is to gain a better understanding of

how the interaction of recruited cytoskeleton proteins as utilized by the actin-based propulsion mechanism of Listeria can produce cellular movements.

Tissue Cell Adhesion

Cells adhere to one another and to the extracellular matrix. These adhesive interactions are mediated by a variety of cell surface receptors. For tissue cell locomotion, it is widely recognized that the integrin receptor family is primarily responsible in providing adhesion and traction to the cell (Ruoslahti and Pierschbacher, 1987; Ruoslahti, 1991). Integrins are large heterodimer transmembrane adhesion proteins that are capable of anchoring the cell to a substrate (Ruoslahti, 1997). Anchorage for the cell is provided through the specific binding of integrins to extracellular matrices, an insoluble meshwork of proteins consisting of collagen, proteoglycan, elastin, hyaluronic acid and glycoproteins (Ruoslahti and Pierschbacher, 1987). In addition, integrins also share an affinity to a number of amino acid sequences, such as arginine-glycine-aspartic acid (RGD), isolated from specific cell binding domains of a number of proteins (Pfaff et al., 1993; Ruoslahti and Pierschbacher, 1986; Ruoslahti, 1996). Long-standing experimental studies have demonstrated the involvement of integrins in cell adhesion (Goodman et al., 1991; Gailit et al., 1997; Akiyama, 1996). More recently, a number of cell migration studies have demonstrated a correlation between cellular movements and integrin-mediated adhesion for protein and protein fragment coated substrates (Stossel, 1993; Palecek et al., 1998; Palecek et al., 1997; Lauffenburger and Horwitz, 1996; Huttenlocher et al., 1995; Huttenlocher et al., 1996).

The Cellular Cytoskeleton

The cellular cytoskeleton is a macromolecular scaffold that allows cells to adopt many shapes and carry out various functions including cellular motion. The cytoskeleton may be considered as an array of highly interconnected filament assemblies (Small et al., 1999)

composed of three distinct filaments: *actin filaments*, *intermediate filaments* and *microtubules* (Pollard and Goldman, 1993; Pollard, 1990). Although each is part of the cytoskeleton, each serves different functions (Table 1-1). In terms of cell motility, actin filaments have been found to be highly associated with the leading edges of cells–the regions most actively involved in tissue cell motility (Welch et al., 1997; Small et al., 1993; Borisy and Svitkina, 2000). Actin filaments typically form networks. Actin networks undergo dynamic reorganization during cell motility via polymerization/depolymerization and the regulated association and dissociation of actin-binding proteins.

Table 1-1 Cellular Cytoskeleton

Cytoskeleton	Function	Reference
Actin filaments	Cell spreading/motility Anchorage to substrate	(Small et al., 1999; Small et al., 1993)
Intermediate filaments	Cell-Cell adhesion Cell-matrix junction	(Small et al., 1999; Pollard and Goldman, 1993)
Microtubules	Vesicle Translocation Transport of Particles/Proteins	(Darnell et al., 1990)

Actin and Actin-Binding Proteins

Actin is the most abundant protein found in non-muscle cells and can exist in the cell in monomeric form (*globular or G-actin*) or as double helical actin filaments (*filamentous or F-actin, or microfilaments*) (Bremer and Aebi, 1992; Pollard, 1990). G-actin is 5.5x5.5x3.5nm in size with a molecular weight of 42kDa (Bremer and Aebi, 1992). Each actin monomer is capable of binding to two other monomers forming a double helical actin filament structure. Actin filaments are regarded as semiflexible biopolymers that give rise to much of the mechanical

properties of the cell. Actin filaments can associate into a number of diverse structures through their interaction with actin-associated proteins. Some actin associated proteins affect the higher order structure (e.g. α-actinin), while others affect the filament polymerization/depolymerization rates (e.g. profilin). In this research, we have focused specifically on the class of actin-bundling proteins that can affect the higher order structure of actin networks.

Actin bundling proteins bind laterally to actin filaments with each bundling protein capable of binding to two actin filaments. The dynamic interaction between actin bundling proteins and actin filaments can result in re-orienting an actin network from randomly oriented filaments to collinearly aligned actin filaments (Figure 1.2). Experimentally this is observed both *in vitro* and *in vivo* (Pavalko and Burridge, 1991; Wong et al., 1983; Wachsstock et al., 1993).

In the intracellular propulsion of Listeria, the actin binding protein α-actinin is found in high concentrations decorated along the lengths of the actin rocket tails (Sechi et al., 1997) and is believed to be an important component in the propulsion mechanism exploited by

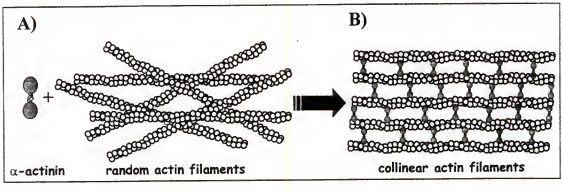

Fig. 1.2. Each α-actinin molecule is capable of forming cross-bridging bonds to actin filaments. In the presence of actin filaments, α-actinin can dynamically reorient a randomly oriented actin network (A) to one described by collinearly aligned actin filaments (B)

Listeria (Dold et al., 1994; Nanavati et al., 1994). Physical and biological binding affinities for α-actinin are given in Table 1-2.

Table 1-2 Alpha-Actinin Biophysical Constants

Property	Biophysical Constant	Reference
Molecular Size	30-40nm x 4-5nm	(Podlubnaya et al., 1975)
Molecular Weight	103kD	(Suzuki et al., 1976)
Head Fragment	27kD	(Bretscher et al., 1979)
Tail Fragment	53kD	(Bretscher et al., 1979)
Binding Affinity		
k_f	$6.8 \times 10^5\ M^{-1}s^{-1}$	(Wachsstock et al., 1994)
k_r	$0.6s^{-1}$	(Wachsstock et al., 1994)
K_d	≈0.6μM	(Wachsstock et al., 1993)
Bond Spacing on F-Actin	≈ $.025nm^{-1}$	(Meyer and Aebi, 1990)

Goals and Objective

Cell locomotion is a very complex and dynamic process involving a number of signaling pathways and molecular components. In this thesis we investigated two areas involved in cell motility.

In the first study we investigated the relationship between three-dimensional cell migration and the adhesiveness of the underlying substratum. Here, we controlled the adhesive properties of a three-dimensional collagen scaffold by covalently grafting adhesive peptides containing the RGD binding sequence. By properly controlling the reaction crosslinking conditions and the number of adhesive peptides grafted to collagen, we could control the adhesive properties of collagen. Using this modified three-dimensional collagen scaffold, we investigated the three-dimensional cell migration of murine melanoma cells as a function of adhesiveness by tracking and recording the time-incremented positions of multiple individual cells in real time. The highlight of this work is the development of a relationship describing three-dimensional cell

migration proficiency as a function of increasing adhesiveness of the collagen scaffold. Building on this study, we then investigated the relationship between cell adhesion and cell migration by developing a three-dimensional RGD concentration gradient functioning to provide a substratum adhesive gradient. The goal for developing this assay was to experimentally test if an adhesive gradient can act as a directional cue in the migration of cells leading to directed migration as predicted by previous theoretical models for adhesion-mediated cell migration (DiMilla et al., 1991; Dickinson and Tranquillo, 1993b). Based on our results, this is the first study, to our knowledge, experimentally demonstrating that adhesive gradients in a three-dimensional scaffold can act as a directional cue for cell migration.

In second study focused on the intracellular motility of Listeria, we investigated the dynamic and regulated interaction of actin filaments and the actin bundling protein α-actinin. We found this interaction can not only result in a transition of randomly oriented filaments to collinearly aligned filaments, but also produce significant cellular forces via an actin filament bundling mechanism. We call this process the "Actin Filament Zipper" (AFZ). Our model is built upon the assumption that dynamic and regulated rearrangements of actin filaments due to cross-bridging interactions between actin and actin-bundling proteins can result in flexural bending of the individual actin filaments during the re-alignment process. The parallel aligning of the actin filaments due to cross-bridging bonds formed between actin filaments can cause a flexural bend in the actin filament(s) potentially resulting in a pushing force. Hence, the chemical energy due specifically to the cross-bridging interaction of actin-bundling proteins energetically drives the AFZ mechanism imparting pushing force. The chemical energy of the system is balanced by the mechanical energy associated with bending an actin filament. In essence, it is the transduction of chemical energy into mechanical energy due to the bundling of actin filaments that is hypothesized as capable of generating significant cellular forces. It is

believed this zippering mechanism can generate significant forces in the propulsion of cells (e.g. *Listeria monocytogenes*).

Two modeling approaches were used to investigate the AFZ mechanism. In the first modeling approach, a thermodynamic model describing the AFZ mechanism is formulated. The reaction kinetics describing the bundling of two actin filaments is modeled as a two-step reversible reaction. The second model additionally accounts for the thermal fluctuations of individual actin filaments. To model these fluctuations, the classical Hookean bead-spring chain model developed by Rouse (Ottinger, 1996) is used. A key result from both models is that the interaction between actin filaments and α-actinin can result in a zippering mechanism of actin filaments and a significant generation of cellular forces.

We also as part of the second study experimentally investigated the motile response of *Listeria monocytogenes* to increasing intracellular concentrations of α-actinin. Intracellular concentrations of α-actinin are elevated by microinjecting purified concentrations α-actinin into Listeria infected cells. To our knowledge this is the first quantitative study relating Listeria speed to intracellular actinin concentration in a dose-response relationship. These experimental results underscore the importance of α-actinin in the intracellular actin-based propulsion of Listeria. Finally, these results additionally lend support to our actin filament zipper hypothesis in that the interaction between actin filaments and α-actinin is a critical to the intracellular propulsion of Listeria and these results qualitatively agree with our theoretical model predictions.

CHAPTER 2
QUANTITATIVE ANALYSIS OF ADHESION-MEDIATED CELL MIGRATION IN THREE-DIMENSIONAL GELS OF RGD-GRAFTED COLLAGEN

Introduction

There is an increasing demand for tissue-engineered biological substitutes as an alternative to transplantation. In some cases, the active migration of tissue cells into and within these implants is critically important to the function of the implant. Because of the established relationship between cell adhesion and cell migration (Huttenlocher et al., 1995; Palecek et al., 1997), one approach to controlling the infiltration or localization of cells is to modify the adhesive properties of the scaffold. Rational design of the adhesive properties of these scaffolds would be greatly aided by a quantitative understanding of the relationship between cell adhesion and cell migration in three-dimensional matrices. Furthermore, in most physiological processes where cell migration is important, such as wound healing, embryogenesis, inflammation, and tumor cell metastasis, cells migrate through three-dimensional tissue matrices rather than on two-dimensional surfaces. Yet, previous quantitative studies of the relationship between cell migration and adhesion have been primarily limited to two-dimensional substrata.

A tissue cell migrates by transmitting cytoplasmic-generated forces into tractional forces to pull itself across the underlying substratum (Sheetz, 1994; Lauffenburger and Horwitz, 1996). Traction is provided, at least in part, by cell-surface adhesion receptors, termed integrins, which bind to complementary adhesion ligands attached to the underlying substratum (Akiyama, 1996). The migration of tissue cells is a complex dynamic process involving a number of discrete events such as receptor-ligand binding, cytoskeleton remodeling, intracellular signaling, force

generation, cell-body displacement, and the rupture of receptor-ligand bonds. The extent or strength of the adhesion between the cell and the underlying extracellular matrix (ECM) is a critical factor in the efficiency of migration (Palecek et al., 1998). Adhesion of the cell to the underlying substratum can be controlled by increasing the adhesion ligand density, receptor ligand binding affinity, or the amount of adhesion receptors expressed on the cell surface (Palecek et al., 1997). Results from quantitative cell-tracking studies of cell migration on two-dimensional surfaces suggest that cell speed is a function of the substratum ligand density (DiMilla et al., 1993; Duband et al., 1991; Huttenlocher et al., 1996; Keely et al., 1995; Palecek et al., 1997). These studies have generally reported that cell speed, is at least in part, regulated through substratum adhesive interactions. The explanation for these observations is that at low adhesiveness, the cell has insufficient traction to propel itself. At high adhesiveness, tractional forces generated by the cell are insufficient to effectively rupture receptor-ligand bonds at the rear of the cell, resulting in an inability of the cell to migrate (Palecek et al., 1998). However, at intermediate adhesiveness, at which receptor-ligand bond strength is on the order of the tractional forces generated by the cell, maximum cell speed is reported.

Although these previous studies have provided much insight into how cells migrate, the relevance of these two-dimensional studies to three-dimensional cell migration through fibrous tissue matrices is unclear because of the limited number of studies using three-dimensional migration assays. Several qualitative differences between two- and three-dimensional migration have been noted (Friedl et al., 1998c; Grinnell, 1982) and suggest the underlying mechanisms may be substantially more complex.

The adhesion and receptor-ligand bond breakage, coupled with the biomechanical resistance imposed on the cell within a three-dimensional fibrous matrix is substantially different than in the two-dimensional environment and requires the cell to respond differently compared to migrating across planar surfaces (Friedl et al., 1998a; Haas et al., 1998). In addition, the cell

migratory behavior within three-dimensional matrices has been reported to vary distinctly among different cell types (Friedl et al., 1998a). For instance, for the highly adhesive integrin-mediated migration of MV3 melanoma cells within three-dimensional collagen matrices, Friedl et al. (1998a) reported that cell migration resulted in a dynamic and permanent reorganization of collagen fibers and receptor shedding. However, it was reported that for the more highly motile integrin-independent migration of T lymphocytes (Friedl et al., 1998b), correlating to a lowered adhesion to the collagen fibrils, T lymphocytes were able to elastically bend the collagen fibers and reduce their cell diameter in order to squeeze through narrow pore spaces within the interwoven fibrous network. Similar results have been observed for dendritic cells migrating within three-dimensional collagen networks (Gunzer et al., 1997). Using a human breast carcinoma cell line, Keely et al. (Keely et al., 1995) compared two-dimensional and three-dimensional migration as a function of cell adhesiveness by reducing the number of $\alpha_2\beta_1$ integrins, a collagen and laminin receptor, expressed by cells. For two-dimensional migration across collagen coated filters, cell migration was optimally supported by intermediate levels of adhesion. In contrast, cells cultured within three-dimensional collagen gels expressing decreased amounts of $\alpha_2\beta_1$ integrins did not organize in three-dimensional collagen gels, suggesting the adhesion and motility was affected differently compared to two-dimensional migration. Further, Kuntz and Saltzman (1997) measured the three-dimensional migration of neutrophils within collagen matrices supplemented with increasing concentrations of ECM proteins preceding collagen gelation. They reported that neutrophil motility was a decreasing function of increasing adhesion and increasing fibril pore spacing indicated through the random motility coefficient, a measure of cell dispersion over time. Together, these studies suggest that the migration of cells within the more highly complex three-dimensional environment may involve alternate biochemical processes and mechanisms to fully elucidate three-dimensional migration within fibrous tissue matrices.

Adhesion and migration is primarily mediated by specific integrins expressed by the cell (Akiyama, 1996). More specifically, adhesion of the cell is mediated through the binding of integrins to discrete regions on ECM proteins (Akiyama, 1996). One ubiquitous adhesive peptide sequence found in many ECM proteins such as fibronectin, laminin and collagen are the arginine-glycine-aspartic-acid (RGD) peptide sequence (Ruoslahti, 1996; Pierschbacher and Ruoslahti, 1984). Many of the integrins share an affinity to the RGD recognition sequence in their binding to ECM ligands (Ruoslahti, 1996). Experimentally this has been established by the treatment of cells with specific RGD-containing peptides to inhibit adhesion and migration of many cell types in response to substratum-bound coatings, implicating the RGD recognition sequence as a mediator of these processes (Alt, 1990).

Recently, Myles et al. (2000) demonstrated that the adhesive properties of collagen could be controlled by the covalent grafting of peptides containing the well-documented cell adhesion amino acid sequence, arginine-glycine-aspartic acid-serine (RGDS), without observable alteration in the gelation time or microstructure of the collagen matrix. The aim of this current work was to quantify three-dimensional cell migration as a function of increasing cell-substratum adhesiveness in reconstituted gels of the covalently modified Type I collagen. Migrating cells were tracked in an automated fashion to generate a large statistical sampling of cell trajectories. The resulting cell migration behavior was quantified by fitting the measured mean squared displacement to predictions from a persistent random walk model using the analysis described in Dickinson and Tranquillo (1993a). This analysis provided estimates of root-mean-square cell speed, directional persistence time, random motility coefficient, and directional persistence length as a function of ligand concentration.

Materials and Methods

Cell Culture

A mouse melanoma cell line (M4 K1735) was generously provided by Dr. Daniel L. Mooradian (University of Minnesota, Departments of Laboratory Medicine and Pathology, Minneapolis, Minnesota). Cells were cultured and frozen in large quantity (cyrogenic storage facilities and assistance in culturing of cell line was graciously provided by the lab of Dr. David Muir at the University of Florida, Department of Pediatrics, Gainesville, FL). The cell line was maintained by *in vitro* culture in low-glucose DMEM with L-glutamine, 110 mg/L sodium pyruvate and pyridoxine hydrochloride supplemented with 10% heat inactivated calf serum and 1.2% penicillin/streptomycin (10,000/10,000 units). Passages were made when cells were 75% confluent using 1X PBS, pH 7.4, and trypsin-EDTA. Cells were discarded after eight passages to minimize phenotypic drift.

Covalent Grafting of RGD/RGE peptides to Type I Collagen

The coupling protocol used to graft glycine-arginine-glycine-aspartic acid-serine-proline-cysteine (GRGDSPC) and arginine-glycine-glutamic acid-serine-proline-cysteine (GRGESPC) has previously been described by Myles et al. (2000). Briefly, reactive groups on both peptides were conserved, thus the reaction chemistry and linking protocol is identical. Type I collagen (Collagen Corporation, Palo Alto, CA) was cross-linked with GRGDSPC/GRGESPC peptide (synthesized by the Interdisciplinary Center for Biotechnology Research, University of Florida, Gainesville, FL) following a two-step reaction involving a heterobifunctional crosslinking reagent, Sulfosuccinimidyl 6-[3'-2-(pyridyldithio)-propionamido-hexanoate (Sulfo-LC-SPDP) (Pierce Biochemical, Rockford, IL). In the first reaction, the N-hydroxysuccinimide reactive group of the Sulfo-LC-SPDP linker was reacted with primary amine groups of collagen

monomers to form stable amide bonds. Excess crosslinker was removed from the derivatized collagen by liquid chromatography (Biorad Econo System). The second reaction involved the 2-pyridyl disulfide reactive group of the Sulfo-LC-SPDP linker to react with the sulfhydryl group present on the cysteine residue of the GRGDSPC/GRGESPC peptide to form disulfide bonds. Conjugated collagen was purified by gel filtration on an 80 mL Sephadex G-200 (Pharmacia Biotech) column eluted with PBS buffer. All reactions were performed in PBS buffer (pH 5) for 24-48 hours at room temperature under constant stirring. Quantification of reactions was completed using spectrophotometric analysis, measuring the absorbance of the pyridine-2-thione leaving group from the crosslinker and measuring the amount of unreacted peptide complexed by Ellman's reagent (Sigma Chemical Company, St. Louis, MO).

Substrate Coating Protocol for Cell Adhesion Experiments

Dilute solutions of 0.24 mg collagen/mL (Collagen Corporation) in 0.5 M glacial acetic acid (Sigma Chemical Co.), were prepared with either unmodified collagen or RGD-grafted collagen (8, 14 and 22 RGD peptides grafted per collagen monomer). First, 0.25 mL of the appropriate collagen solution was added to the wells of a 24 well Corning cell culture plate. The wells were incubated at room temperature for one hour to allow protein adsorption to wells. Next wells were rinsed with several milliliters of PBS buffer to remove any excess acetic acid. To block any exposed polystyrene, 0.25 mL of a 0.5% w/v solution of bovine serum albumin (BSA) (Sigma Chemical Co.), was then added to each well and incubated for one hour at room temperature. Each well was gently rinsed again with several milliliters of PBS to wash away any non-adherent protein. As control, a set of wells were coated with BSA and PBS (uncoated polystyrene surface). Each condition was duplicated in 4 wells.

Automated Video Microscopy Incubation System

Cell motility and adhesion experiments were performed using a Nikon Diaphot 200 inverted microscope equipped with computer-controlled motorized stage and focus for automated x, y, z-positioning (Ludl Electronics, Pleasantville, New York). Images were captured using a CCD camera and image processing software Optimas™ (Media Cybernetics, Silver Spring, MD). To maintain cells at physiological conditions during experiments, a custom-designed stage incubation system was built which mounts onto the microscope body maintaining temperature control (37°C), high humidity level (relative humidity>85%) and a 5% carbon dioxide atmosphere. The atmosphere inside incubation system was well mixed using two 4" circular fans to prevent any unevenly heated regions inside incubation system. Physiological conditions were maintained and monitored inside the incubation system during experiments. using a data acquisition board (Keithley DAS 1801 ST-DA, Cleveland, OH). A data acquisition program was written to record temperature, carbon dioxide and humidity levels inside the incubation system.

Adhesion Experiment Methodology

Cells were harvested and re-suspended in serum-free culture media at 50,000 cells/mL. A cell suspension of 0.5 mL was added to each well and incubated for 30 minutes to allow cells to attach to the substrate. Samples were then transferred to the stage microscope incubation system which was maintained at 37°C, 5% CO_2, and relative humidity >85% during the experiments. Using a computer program written in Optimas™ programming language (Analytical Language for Images), the motorized stage scanned each tissue culture well in an automated fashion to capture and process images and determine cell counts in each microscope field. After the initial cell counts, samples were then placed on a Rotomix (Barnstead/Thermolyne, Dubuque, Iowa) at 100 RPM (inside incubation system) and spun to

remove approximately 50% of attached cells on the control (unconjugated) collagen. Next, detached cells were removed from each well by gently aspirating the media. Finally, cells were fixed using neutral buffered formalin (10%) and recounted using the computer program described above to determine final cell density per microscope field.

Preparation of Cell Migration Assay

Migration of murine melanoma cells was quantified within gels of unmodified collagen and GRGDSPC/GRGESPC-grafted collagen. Collagen gels were prepared with the following composition (volume basis): 67% collagen solution, 13% 10X DMEM, 5% 4.4% w/v sodium bicarbonate, 10% 0.1M NaOH, and 5% cell solution (cells suspended in serum-free DMEM). Solutions were well mixed to insure uniform dispersal of cells in the collagen solutions (10^4 cells/mL) and pH to 7.2. Collagen solutions were heat polymerized in a cell culture incubator (37°C, 5% CO_2, relative humidity >98%) for two hours. Upon polymerization of the collagen gels, 0.2 mL of serum-free culture medium was added to each sample to provide nutrients for cells during tracking experiments. The samples were then transferred to the pre-heated and humidified microscopy incubation system to begin a cell tracking experiment.

Three-Dimensional Cell Tracking

For direct observation of cell migration, the positions of multiple individual cells were tracked in real time using the automated video microscopy system. The tracking methodology has previously been described (Dickinson et al., 1994). Briefly, a computer program was written to drive the motorized stage and focus control and process digitized images in order to track the positions of individual cells in an automated fashion. To accomplish this, the initial three-dimensional positions of the individual cells to be tracked were found manually using the motorized stage and recorded to an output file. After locating the initial positions of all cells to

be tracked, the computer algorithm recursively returned to the stored position of cells and updated the three-dimensional position of each cell. First, the focal plane was scanned to determine the plane of highest optical contrast, which was set as the new z-coordinate of the cell. Next, a series of image processing steps was completed to enhance the image cell cortex and determine the projected cell centroid, which was set as the new x, y-coordinate for the cell. Finally, the microscope stage was re-centered such that the cell's center of mass was in the center of the projected image and the updated position of the cell was recorded to an output file. This was recursively executed for all cells and all time points. For each experiment involving RGD-grafted collagen, 40 cells were tracked. The time interval between measurements for each cell was approximately 16 minutes. For experiments involving RGE-grafted collagen, 80 cells were tracked to increase statistical confidence while lowering the number of repeat experiments. The time increment was approximately 35 minutes for these experiments. All tracking experiments were repeated at least two times. Cells were tracked for approximately 15 hours in each experiment.

Statistical Analysis of Cell Paths

A number of statistics were generated from the cell paths to quantify the time-average and population-average migration behavior for each sample. The mean-squared displacement vs. time was estimated by averaging the squared displacement over non-overlapping time intervals along each cell path, then pooling the average among the population of tracked cells. Using the statistical analysis previously described by Dickinson and Tranquillo (Dickinson and Tranquillo,1993a), the theoretical mean-squared displacement, $<d^2(t)>$, vs. time, t, based on a persistent random walk model (Dunn, 1983), was fit to the experimental data, using generalized nonlinear least squares regression. The fitted equation has the following form:

$$< d^2(t) >= 2n_d\mu(t - P(1 - e^{-t/P})) + 2\gamma \qquad (2.1)$$

where n_d is the number of dimensions in which the data analyzed, μ is the *random motility coefficient*, t is time, P is the *directional persistence time*, and γ is the *mean square positioning error* (Dickinson and Tranquillo, 1993a). The *random motility coefficient* is a measure of the rate of cell dispersion over longer time due to the random migration behavior of the cells. The *directional persistence time* is a measure of the mean time a cell persists in the same direction before random turning results in a loss in directional memory. The root-mean-square cell speed is related to μ and P by the following relationship (Dunn, 1983):

$$S = \sqrt{\frac{\mu n_d}{P}} \tag{2.2}$$

Another measure of directional persistence is the *directional persistence length*, P_L (Alt, 1990), defined as $P_L=SP$, and is the characteristic distance traveled by the cells in the same direction.

Note that the choice of the tracking time increment, Δt, influences the precision in the estimates of μ, P, and S. As shown in Dickinson and Tranquillo (Dickinson and Tranquillo, 1993a), a time increment much larger than about $5P$ can result in large standard errors in the estimates of P and S. On the other hand, a small time increment allows fewer cells to be tracked per experiment. Dickinson and Tranquillo (Dickinson and Tranquillo, 1993a) showed that P and S can be estimated with good precision when the tracking time increment is close to P. Based on an *a priori* estimate of P, a time increment of approximately 15 minutes was chosen to maximize the number of cells tracked at minimal cost in precision in the estimates of S and P.

Results

Cell-Substratum Adhesion Analysis

The adhesive peptide chosen to graft to collagen was glycine-arginine-glycine-aspartic acid serine-proline-cysteine (GRGDSPC, hereafter referred to as RGD) peptide, containing the

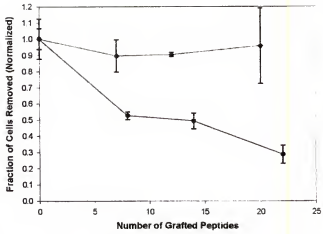

Fig 2.1. Effect of substratum adhesiveness as a function of grafted peptides conjugated to collagen. Data for GRGDSPC (•) and GRGEDSPC (♦) peptides conjugated to collagen. Values are normalized by cell adhesion on collagen. Error bars represent 95% confidence intervals based on mean of samples.

well-documented adhesive RGD peptide sequence (Pierschbacher et al., 1985; Pierschbacher and Ruoslahti, 1984; Ruoslahti, 1996; Ruoslahti et al., 1987). As an experimental control, a nonadhesive peptide composed of glycine-arginine-glycine-glutamic acid serine-proline-cysteine (GRGESPC, hereafter referred to as RGE) was also grafted to collagen (Dedhar et al., 1987; Pierschbacher and Ruoslahti, 1984). This was done to determine whether differences between collagen and RGD-grafted collagen was due to differing adhesiveness or from the grafting process itself.

Cell-substratum adhesiveness was measured as a function of increasing number of RGD and RGE peptides conjugated to collagen. Preliminary cell adhesion tests were performed to determine if K1735 M4 murine cells expressed specific receptors for the adhesive RGD peptide chosen. This was accomplished by covalently grafting the RGD peptide to a nonadhesive blood protein, bovine serum albumin (BSA) as previously described (Myles et al., 2000). Cell adhesion to RGD-grafted BSA increased significantly compared to unmodified BSA, suggesting murine cells were able to adhere directly to the adhesive peptide (Myles et al., 2000).

Further, to determine if the grafting of RGD peptides to collagen affected cell-substratum adhesiveness multiple cell adhesion experiments involving collagen and increasing amounts of RGD peptides conjugated to collagen were performed. As shown in Figure 2.1, the number of

detached cells decreased with increasing degree of peptide conjugation. The adhesion of cells to the RGE-grafted collagen substrates was not found to depend on the amount of RGE peptides present, nor was it significantly different from that of unmodified collagen (Fig. 2.1). These data suggest that the enhanced adhesion was specifically due to cell binding to the grafted RGD, rather than a secondary effect due to artifacts of the grafting procedure.

Cell Migration Results

Multiple cell tracking experiments were performed within three-dimensional collagen matrices as a function of increasing degree of RGD and RGE conjugation to collagen. From the cell tracking experiments, the squared displacement of the cells as function of increasing time was computed from the time-incremented positions of cells by applying non-overlapping time intervals analysis as previously described (Dickinson and Tranquillo, 1993a). To maximize the available data for each time interval, multiple cell tracks were averaged to estimate the average migration of the cell population, pooled mean-squared displacement, *(this assumes the cell population behaves sufficiently homogenous)* over all cell

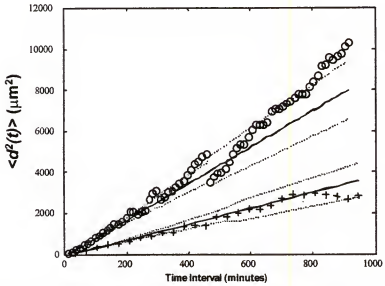

Fig. 2.2. Pooled mean-squared displacement computed from time-incremented cell positions. The solid line represents fitted line to the model and dashed lines represent $\langle d^2(t)\rangle \pm \sigma$ (expected standard deviation). Collagen grafted with 8 (O) (60 cells) and 22 (+) RGD peptides (66 cells) grafted shown.

tracks. Since the distance calculated for increasing time intervals is not statistically independent (Dickinson and Tranquillo, 1993a), the pooled mean-squared displacement data appears to be correlated as shown in Figure 2.2. For this reason, to obtain optimal estimates of the parameters from the persistent random walk model, a generalized nonlinear least squares regression

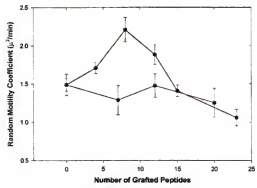

Fig. 2.3. Relationship between random motility coefficient (μm^2/min), a measure of cell dispersion, as a function of increasing GRGDSPC (•) and GRGESPC (♦) peptides conjugated to collagen. Error bars represent ± standard error.

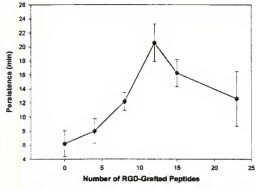

Fig. 2.4. Relationship between persistence time, characteristic time cells persist without significant directional change (min), as a function of increasing GRGDSPC peptides conjugated to collagen. Error bars represent ± standard error.

procedure was used to fit the model to cell tracks, which accounts for the fact that the residuals are not independent and identically distributed because less data is available at larger time intervals (Dickinson and Tranquillo, 1993a). The generalized nonlinear regression analysis calculates the theoretical variance-covariance matrix of the residuals to properly weight the residuals and account for correlation (Dickinson and Tranquillo, 1993a). To illustrate the fitting procedure, Figure 2.2 shows a fit of equation (1) to the pooled mean-squared displacement from cells tracked within gels containing 8 and 22 RGD peptides grafted per collagen monomer.

The reported parameter estimates reflect average quantities over the course of cell tracking experiments. To insure gels were uniform, cell tracks were analyzed to determine if

there was a bias in the migration towards the x, y or z direction. No consistent directional migration bias was observed (data not shown), which suggests that the collagen gels were isotropic. Furthermore, to determine whether the measured cell migration statistics were constant in time, cell-tracking data sets were divided into halves corresponding to the earlier and later halves of the total tracking time. No significant statistical difference was found between cell migration parameters resulting from data taken from the two halves (data not shown), suggesting that, at least over the time scale of these experiments, the cell migration behavior did not change greatly over the course of the experiments.

The random motility coefficient, μ, was found to have a biphasic dependence on cell-substratum adhesiveness with a maximum increase at intermediate adhesiveness (Fig.2.3). No significant differences were found between the random motility coefficient for the unmodified collagen compared to the RGE-grafted collagen. This suggests that the alteration of the cell migration was specifically due to the presence of RGD, not the alteration of some other gel properties due to the grafting procedure.

To further investigate the differences that might contribute to the biphasic dependence of the random motility coefficient on the degree of RGD-grafting to collagen, both the directional persistence time and cell speed, which contribute, to the random motility coefficient were examined. The directional persistence time, P, is inversely related to the frequency of directional changes. P was measured to have a biphasic dependence on the degree of RGD-conjugation as shown in Figure 2.4. However, the root-mean square cell speed, S, calculated from Eqn. 2, was found to monotonically decrease with increasing RGD peptide conjugation (Fig.2.5). No significant dependencies were found for RGE-grafted collagen. These data suggest that

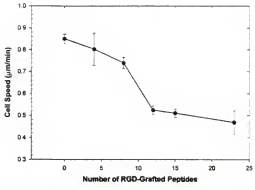

Fig. 2.5. Relationship between cell speed (μm/min) as a function of increasing GRGDSPC peptides conjugated to collagen. Error bars represent ± standard error.

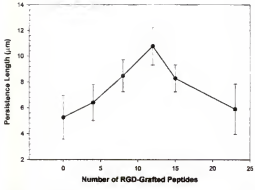

Fig. 2.6. Relationship between persistence length, characteristic length cells migrate without significant directional change (μm), as a function of increasing GRGDSPC peptides conjugated to collagen. Error bars represent ± standard error.

increasing substratum adhesiveness reduced cell speed, but there was an optimal level of adhesiveness for enhanced directional persistence in the collagen gels.

The persistence length, P_L, is a measure of the tortuosity of the cell path. P_L also showed a maximum with respect to the degree of RGD conjugation (Fig.2.6). Therefore, optimal levels of adhesiveness were observed for both the length of time and the distance cells persist in the same direction.

Discussion

The migration of a cell within a three-dimensional fibrous network is a highly dynamic and complicated process requiring the cell not only to properly adhere to the substratum, generate tractional forces to propel itself and rupture existing cell-substrate adhesions at the rear of the cell, but also to overcome the biomechanical resistance imposed on the cell due to the interwoven

fibrous network of collagen fibrils. Recent cell migration studies within three-dimensional collagen matrices have found that regulation of cell migration and the migratory response can vary substantially depending on the cell type (Friedl et al., 1998b; Friedl et al., 1997; Friedl et al., 1998c). In addition, the up-regulation of enzymatic proteins such as metalloproteinases within three-dimensional cultures compared to two-dimensional cultures has been reported to be essential in the proper migration and cell organization for endothelial and human tumor cells within three-dimensional collagen matrices (Deryugina et al., 1998; Haas et al., 1998). Although the migration of a cell is very complex by nature, it is clear that proper cell-substratum adherence is critically important for proficient cell migration. The aim of this study was to examine three-dimensional cell migration within collagen matrices as a function of increasing cell-substratum adhesiveness through the grafting of an adhesive peptide to collagen using a well-established murine melanoma cell line.

Utilizing the coupling procedure developed by Myles et al. (2000) the grafting of adhesive RGD peptides to the collagen prior to fibrillogenesis provided a method to systematically amplify the adhesive properties of Type I collagen. One concern was that the increased adhesiveness might be attributed to a changed confirmation structure of the protein itself (Morla et al., 1994) instead of the grafted adhesive peptide. To investigate this, a peptide, which was both similar in size and structure with the only difference being attributed to the deletion of aspartic acid and insertion of glutamic acid, that has been reported as a nonadhesive peptide for many cell types (Dedhar et al., 1987; Pierschbacher and Ruoslahti, 1984) was used as a control. Based on the adhesion studies, comparing the adhesive peptide to the nonadhesive perturbing peptide grafted to collagen, suggests the increased adhesive properties were specific to the adhesive RGD-grafted peptide rather than changes due to the grafting process itself, such as conformational changes in protein configuration.

In addition to the adhesive characteristics of the substratum, it has previously been reported that the physical structure of the substratum can also affect cell migration (Dickinson et al., 1994; Kuntz and Saltzman, 1997). More specifically, both the fibril pore size and fibril arrangement can affect cell motility within collagen gels. Kuntz and Saltzman (1997) reported the migration of neutrophils within collagen gels was a decreasing function of increasing pore size, while Dickinson et al. (1994) reported that for oriented collagen fibrils, biased migration was observed along the major axis of fibril orientation. To investigate this, scanning electron microscopy was used to probe the microstructure as a function of increasing RGD concentration. No significant differences in average pore size or fibril diameter were found between the unconjugated and conjugated collagen gels (Myles et al., 2000). This further suggests differences in cell motility within collagen gels were specifically due to a differential in substratum adhesiveness rather than the physical structure of the gel.

Migration of melanoma cells is integrin-dependent, likely involving multiple cell-surface receptors. As previously reported, melanoma cells express both $\alpha_1\beta_1$ (VLA-1) and $\alpha_2\beta_1$ (VLA-2), both which have a high affinity for collagen (Kramer and Marks, 1989). Although the Type I collagen sequence contains multiple RGD amino acid sequences, the $\alpha_1\beta_1$ and $\alpha_2\beta_1$ receptors have previously been shown to be RGD-concentration independent for native collagen (Berman et al., 1993). This suggests, at least a third cell surface integrin receptor is likely involved in the adhesion of these cells to the grafted RGD peptides. Based on a literature review, the most likely integrin responsible for binding to the grafted-RGD in a concentration dependent manner, is the $\alpha_v\beta_3$ integrin expressed by melanoma cells (Aznavoorian et al., 1996; Morla et al., 1994). Other integrins expressed by melanoma cells have previously been reported (Albelda et al., 1990; Kramer et al., 1991). The involvement of these integrins in cell migration for Type I collagen has been addressed previously (Keely et al., 1995). However, the extent and involvement of these integrins in the migration through the conjugated and unconjugated collagen gels was not

investigated here and could be the subject of future work. It should be pointed out that, the involvement of multiple integrins could activate multiple signaling pathways affecting the motile processes of the cell and thus could be potentially responsible for some of the motile behaviors observed in this study compared to others.

In contrast to a biphasic dependence for cell speed as a function of increasing cell-substratum adhesiveness as reported for several two-dimensional migration studies (DiMilla et al., 1993; Duband et al., 1991; Huttenlocher et al., 1996; Keely et al., 1995; Palecek et al., 1997), our studies suggest the migration of melanoma cells within collagen gels was a monotonically decreasing function of increasing adhesiveness. Possible reasons for such discrepancies reported between this study and others previously referred to could be attributed to the different cell types used between these studies. Other significant differences between these studies and the current one should also be mentioned. Melanoma cells form strong stable adhesions to the collagen fibrils as exhibited by the high degree of spreading and asymmetry. Further illustrating this, recently Friedl et al. (1997) observed that for MV3 melanoma cells migrating within Type I collagen matrices a substantial fraction of the α_2 and β_1-integrins were released at the site of detachment remaining associated with the substratum, correlating to a high cell-substratum adhesiveness (Palecek et al., 1998). The grafting of additional adhesive RGD peptides to the collagen fibrils is expected to further amplify the adhesiveness of the collagen substratum, thus resulting in a limited range of adhesiveness we could explore. Consequently, the least adhesive collagen matrices (unconjugated collagen) could compare to a substantially high degree of adhesiveness compared to other studies and therefore result in a position on the predicted biphasic cell speed diagram (DiMilla et al., 1993) beyond the maximal cell speed at which point cell speed then becomes a decreasing function of increasing adhesiveness. Another possible explanation for these differences could be that for a given level of adhesiveness, the total *effective* adhesion a cell experiences within a three-dimensional matrix, coupled with the biomechanical resistance of the

fibril network imposed on the cell itself is considerably greater than for two-dimensional substratum. Thus, points of lower substratum adhesiveness could be functionally shifted to the right along on the biphasic curve, resulting in consistent trends as seen in previous two-dimensional studies.

Cell persistence time and persistence length did exhibit a biphasic dependence on increasing cell-substratum adhesion, suggesting cells migrated over longer distances without significant directional changes at intermediate adhesiveness. Although most previous studies for two-dimensional migration have reported instead a biphasic dependence of cell speed as a function of cell-substratum adhesiveness, it is worth noting Bergman and Zygourakis (1998) recently observed that with increasing concentrations of surfaces coated with fibronectin, using a lymphocyte cell line, enhanced cell migration was correlated to increased persistence time with no significant changes in cell speed. Friedl et al. (1998c) reported that migration of MV3 melanoma cell migration within collagen matrices resulted in a characteristic reorganization of the surrounding collagen matrix, leading to the alignment of collagen fibers at binding sites, resulting in a contact guidance-like mechanism. As a result, cells migrated along a pre-existing pathway with a high directional persistence. Another possibility is that increasing cell-substratum adhesiveness resulted in the cell becoming hyper-adherent to the substrate with competing pseudopods in multiple directions resulting in decreased persistence time.

This is the first quantitative study of three-dimensional migration as a function of increasing substratum adhesiveness using an adhesive peptide grafted to collagen. Our results for the three-dimensional migration within collagen matrices tend to agree with previous work in collagen gels (Aznavoorian et al., 1996) in the sense that migration of melanoma cells is more directly correlated with persistence length than cell speed. Current and future work include studying three-dimensional cell migration within spatial adhesive gradients to determine if biased migration (*haptotaxis*) occurs and further understanding the involvement of β_1 and β_3-integrins in

binding to the substratum and migration.

CHAPTER 3
ADHESIVE GRADIENT: A POTENTIAL DIRECTIONAL CUE FOR CELL MIGRATION

Introduction

There are a number medical implant devices such as synthetic vascular grafts, synthetic skin grafts and cartilage scaffold replacements that rely heavily on the incorporation of normal functioning tissue cells on and into the implant device itself. One attractive strategy to address this issue is through directed migration of native surrounding tissue cells to targeted therapeutic sites. One potentially promising approach to invoke directionally biased migration is through the use of a gradient of substratum-bound adhesion molecules (*haptotaxis*). To date, however, there have been limited quantitative studies reported for cells exposed to gradients of substratum-bound adhesion molecules. Consequently, a better fundamental understanding of haptotaxis migration within a three-dimensional environment typical of physiological conditions is required before such strategies can be most successfully utilized in therapeutic applications.

An actively migrating tissue cell is normally polarized extending and retracting pseudopods primarily in the direction of movement. For a tissue cell migrating within an isotropic environment, the cell tends to retain a "directional" memory over short periods of time, persisting in the same direction without significant directional changes. For longer time periods (much longer than its directional memory), the cell tends to loose its directional memory and its cell path is characterized by more of a meandering path, consistent with molecular diffusion processes. This type of migration is termed "random motility" (Dickinson and Tranquillo, 1993b). In contrast, when a cell encounters an anisotropic environment, such as a concentration gradient of a cell-stimulus, its path can be modified, resulting in directional-biased migration.

Cellular movements may be modified in a number of possible ways to elicit a potential biased response. Accordingly, a number of modes of migration have been proposed (Dickinson and Tranquillo, 1993b). For example, a cell may preferentially turn towards or away from the direction of increasing cell-stimulus, a phenomena termed topotaxis (Dickinson and Tranquillo, 1993b). Another possibility is a dependence of cell speed and/or turning frequency on the direction of movement, a phenomena termed orthotaxis and klinotaxis respectively (Wilkinson et al., 1984; Schnitzer and Juliano, 1990). Also, a dependence of cell speed and turning behavior on the magnitude of cell-stimulus can result in directionally-biased migration, referred to as orthokinesis and klinokinesis respectively (Tranquillo and Alt, 1990).

The migration of a cell in response to a substratum-bound gradient has been proposed as a mechanism involved in a number of physiological processes such as wound inflammation (Rich and Harris, 1981), embryogenesis (Erickson, 1985) and neurite outgrowth (Buettner et al., 1994). The concept haptotaxis was first proposed by Carter (1965) in which he described cellular movements to be controlled by the distribution of relative strengths of cell adhesion along the cell body. Later, Harris confirmed Carter's earlier observations using substrata composed of a continuous non-biologic adhesion gradient noting that the cells did moved up and down the gradient, but tended to accumulate in regions of increased adhesiveness (Harris, 1973). Harris hypothesized that the leading lamella attached more firmly in areas of higher adhesiveness, thus creating a sort of tug-of-war between competing the front and rear of the cell; with the more firmly attached front eventually winning out and the cell propelling forward. Still, others have proposed that haptotaxis may be a special case of chemotaxis (a biased migration response due to a soluble chemotactic agent) (Keller et al., 1979). However, more recently it has become increasingly clear based on a number of studies that suggest cells utilize separate and different signaling pathway for chemotaxis (G-protein mediated pathway) and haptotaxis (integrin-mediated pathway) (Hauzenberger et al., 1994; Klominek et al., 1997; Aznavoorian et al., 1990).

More recently in supporting of Harris's hypothesis, Palecek et al. (1998), reported a similar mechanism of competing lamella attachment and rear-detachment in the migration of Chinese hamster ovary cells for protein-coated surfaces.

Since Carter and Harris's early reports of haptotaxis, a number studies have reported similar observations using filter-based assays consisting of a porous filter which is soaked in a protein containing solution on side and the other side of the filter is exposed to a protein-free solution. Thus, due the presence of a protein concentration gradient across the filter, presumably an gradient of adhesive protein is adsorbed to the filter. (Hauzenberger et al., 1994; Klominek et al., 1997; McCarthy et al., 1983; McCarthy and Furcht, 1984). However, potential problems associated with these assays are noteworthy. Little, if any, consideration was given to the possibility of protein desorption from the protein coated filters, and it is likely that a portion of the absorbed molecules desorb during the course of the experiment, perhaps contaminating the results with a chemotactic response (Grinnell, 1982). Further, direct observation of cell migration is impossible using this experimental design thus providing little information regarding actual cell displacement, consequently, limiting the interpretation of the data relating cellular movements to the presence of an adhesive gradient.

Aimed on improving the controllability of the adhesive gradient, Brandley and Schnaar (1989) created immobilized gradients of RGD-containing peptides on polyacrylamide gel surfaces affording the advantage by covalently immobilizing adhesion molecules to the substratum. Brandley and Schnaar reported melanoma cells redistributed towards areas of higher immobilized peptide densities while cells on uniform concentration of adhesive peptide gels did not redistribute. Their conclusion to these observations was that the redistribution of cells was specifically due to haptotaxis migration. Though this assay offers improvement in the controllability and the properties of the adhesive gradient, some pitfalls to this design and caution suggested and in the interpretation of their analysis should be pointed out. Based on the plots

illustrating cell density along the length of the gradient, cell density is decreased along the edges of the gel, suggesting the washing process used in their protocol removed excessive amounts of cells particularly at the edges where the shear stress would be the greatest. This additionally opens the potential that the washing process also removed a substantial number of cells on the lesser adhesive region of the substratum. If this were true, this would potentially explain why they observed higher cell densities on regions of higher adhesiveness. Since individual cell displacements during the course of the experiment were not recorded, one cannot ascertain, based on the results presented, whether the cells actually redistributed due to haptotaxis migration or was increased cell density on the higher immobilized peptide regions as a result of increased adhesiveness.

Although there are a number of studies supporting haptotaxis migration, the experimental evidence supporting this mechanism is weak at best and to date, no experimental study to date has demonstrated haptotaxis migration through direct observation on a continuous gradient of biological adhesion molecules. The aim of this current work was three-fold: to study 1) three-dimensional cell migration in a biologically relevant cell-matrix, 2) develop a method to control adhesive gradient steepness and 3) directly observe, and characterize the time-incremented cell displacements in response to the adhesive gradients.

Recently Myles et al. (2000), demonstrated that the adhesive properties of collagen could be controlled by covalent grafting adhesion peptides containing the well-documented amino acid sequence, arginine-glycine-aspartic acid serine (RGDS). This assay was demonstrated to have no observable differences in the gelation time or microstructure of the collagen matrix. Using this collagen assay, Burgess et al. (2000), reported three-dimensional melanoma cell migration was a biphasic function of increasing substratum adhesiveness. These results suggested, along with other studies (Palecek et al., 1997; Duband et al., 1991; Huttenlocher et al., 1996; DiMilla et al.,

1993), an optimum for substratum adhesion exists for enhanced cell migration, thus raising the question whether a gradient of adhesiveness could be utilized to direct cell migration.

Using the modified collagen assay developed by Myles et al. (2000), a procedure for constructing three-dimensional haptotaxis assay with RGD-concentration gradients was developed. This procedure provided a three-dimensional collagen scaffold 1) with cells interspersed within the matrix, 2) optically could be viewed with a microscope, and 3) a method to control the adhesive gradient steepness. Using the haptotaxis assay, an investigation of cell migration as a function of increasing gradient steepness was completed. Recording the time-incremented positions of cells as a function of increasing adhesive gradient, a large statistical sampling of cell trajectories was generated. The resulting cell migration behavior was mathematically analyzed to determine if, and to what extend, the cells' paths were modified by the presence of an adhesive gradient. The analysis provided estimates for root-mean cell speed, average drift velocity, cell orientation bias, and haptotaxis coefficient as a function of gradient steepness.

Materials and Methods

Cell Culture

A highly metastatic clone (M4) of the K1735 mouse melanoma line was provided by Dr. Daniel L. Mooradian (University of Minnesota, Departments of Laboratory Medicine and Pathology, Minneapolis, MN). The cell line was maintained as previously described (Burgess et al., 2000). Cell passages were made at 70-80% confluency by washing twice with 5 ml of 1X PBS, pH 7.4, releasing with 2 ml of trypsin-EDTA and then replating 1:5 in low-glucose DMEM with serum. The number of passages was limited to 8 in order to minimize phenotypic drift of the cell population. Cells were harvested for use in experiments by the same procedure used for

passage, however, they were resuspended in serum free DMEM. All materials for cell culture were purchased from Gibco BRL, Grand Island, NY.

Grafting of GRGDSPC/GRGESPC Peptides to Collagen

The coupling protocol used to graft glycine-arginine-glycine-aspartic acid-serine-proline-cysteine (GRGDSPC) and arginine-glycine-glutamic acid-serine-proline-cysteine (GRGESPC) peptides to collagen has previously been described by Myles et al. (2000). Briefly, type I collagen (Collagen Corporation, Palo Alto, CA) was coupled with GRGDSPC or GRGESPC peptide (synthesized by the Interdisciplinary Center for Biotechnology Research, University of Florida, Gainesville, FL) via the heterobifunctional reagent Sulfosuccinimidyl 6-[3'-2-(pyridyldithio)-propionamido]hexanoate (Sulfo-LC-SPDP) (Pierce Biochemical, Rockford, IL). The procedure involved a two step reaction with purification and concentration after each reaction. In the first reaction, the N-hydroxysuccinimide group of the Sulfo-LC-SPDP linker was reacted with primary amine groups of collagen monomers to form stable amide bonds and create a derivatized collagen. Excess Sulfo-LC-SPDP was removed from the derivatized collagen by liquid chromatography on the Biorad Econo System (Biorad Laboratories, Hercules, CA) using a desalting column and concentrated using Centricon-30 micro-concentrators (Amicon Corporation, Beverly, MA). In the second reaction, the 2-pyridyl disulfide group of the Sulfo-LC-SPDP reagent on the derivatized collagen was reacted with the sulfhydryl group on the cysteine residue of the GRGDSPC or GRGESPC peptide to form disulfide bonds and create the collagen-RGD or collagen-RGE conjugate. The unreacted peptide was removed from the conjugated collagen mixture by gel filtration liquid chromatography and concentrated using the Centricon-30 micro-concentrators. All reactions were performed in PBS buffer (pH 5) for 24-48 hours at room temperature under constant stirring. Quantification of reactions was completed by spectrophotometric analysis, measuring the absorbance of the pyridine-2-thione leaving group

from the Sulfo-LC-SPDP reagent and als by measuring the amount of unreacted peptide complexed by Ellman's reagent (Sigma Chemical Company, St. Louis, MO).

Automated Video Microscopy Incubation System

Cell motility experiments were performed using a Nikon Diaphot 200 inverted microscope equipped with computer-controlled motorized stage and focus for automated x, y, z-positioning (Ludl Electronics, Pleasantville, New York) as previously described (Burgess et al., 2000). Briefly, images were captured using a CCD camera and image processing software Optimas™ (Media Cybernetics, Silver Springs, MD). Using a custom-designed stage incubation system that mounted onto the microscope body, cells were maintained at physiological conditions (temperature control 37°C, humidity level >85% relative humidity, 5% carbon dioxide atmosphere. Using a data acquisition board (Keithley DAS 1801 ST-DA, Cleveland, OH), physiological conditions were maintained and monitored.

Adhesive Gradient Chamber Design

Cell motility experiments were completed using a custom-designed adhesive gradient chamber. The chamber was constructed using two 25 x 75mm standard microscope glass slides mounted between a 23 x 17 x 5mm Teflon spacer (Fig. 3.1). The glass slides were glued to the Teflon spacer using a silicone rubber adhesive and allowed to dry for 24 hours. At the bottom of the Teflon chamber a small opening (diameter ≈3.5mm) was drilled to allow the to pumping collagen solutions into the chamber. The gradient chamber was sterilized using an autoclave prior to each experiment. On the day of an experiment, the top of the chamber was sealed using wax. Just prior to the chamber being filled with the collagen solutions, a small hole in the wax seal at the top of the chamber was created to allow the air within the chamber to be displaced while pumping the collagen solutions into the chamber. During the pumping procedure and the

heat polymerization steps, the chamber remained fixed in a vertical position as shown in Figure 3.1. Following each tracking experiment, the chamber was disassembled and cleaned thoroughly.

Preparation of Adhesive Gradient Migration Assay

Adhesive gradients were created in an automated fashion using the adhesive gradient chamber (described above) and computer controlled peristaltic pump equipped with a gradient former (BioRad Econo liquid chromatography system). This unit could control the relative amounts (*volume basis*) of two solutions, *A* and *B* being pumped and could be programmed to pour varying gradient steepnesses by programming the proper flow rates and relative amounts of solution *A* and *B* desired at various times during the pumping cycle. To create the adhesive gradients in these experiments, solution *A* was chosen to be collagen (unmodified collagen) and solution *B* RGD-grafted collagen. By varying the volumetric flow rate ratios of each solution as a function of time, an RGD-gradient could be created. Substituting solution B for an RGD-grafted collagen solution with a higher number of RGD peptides grafted to collagen provided a means to increase RGD-gradient steepness. Collagen solutions *A* and *B* were prepared using protocol as previously described (Burgess et al., 2000). Once prepared, collagen solutions were well mixed to insure uniform dispersal of cells in collagen solutions, brought to a final pH of 7.2, and loaded onto the gradient pump unit. Final cell concentration within each solution was 50,000 cells/ml.

Prior to experiments all tubing was sterilized using an autoclave. Preceding the pumping of solutions *A* and *B*, a PBS buffer solution followed by 3ml of unmodified collagen solution was first pumped through tubing system and collected in a waste bin. The PBS buffer provided a wash facilitating the removal of any foreign deposits that might be on the interior walls of the tubing and the collagen solution provided a protein-coated surface adsorbed to the inner walls of the tubing. Both steps were found to be critical in the successful formation of adhesive gradients.

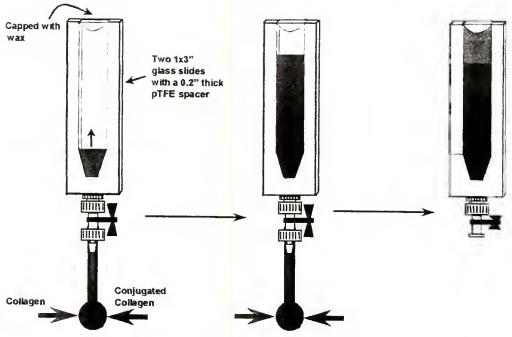

Fig. 3.1. Schematic illustrating the procedure for formation of spatial adhesion gradients. To minimize dispersion, collagen solutions were pumped from the bottom. Volumetric ratio of modified/unmodified collagen solution was changed linearly a linear gradient of the two solutions. Once the gradient was pumped, the chamber was carefully placed in a vertical rack within a cell culture incubator and heat polymerized (T=37°C, 5% CO_2) for two hours.

Once the tubes were washed and protein coated, the gradient pump unit was programmed to pump 1 ml of solution *A* (100% solution *A*), linear gradient of solution *A* and *B* (0% to 100% solution *B*, total volume 1ml) followed by 1 ml of solution *B* (100% solution *B*). Pumped solutions were pumped at flow rate of 0.25ml/min into the bottom of the gradient chamber. Once pumping was complete, the bottom of the chamber was sealed to prevent any leakage of solutions from bottom of chamber. Next, chamber was transferred to a storage rack and heat polymerized within a cell culture incubation system (37°C, 5% CO_2, high relative humidity) for two hours. Following polymerization, collagen gels were supplemented with DMEM to provide nutrients to cells during the course of the experiment. DMEM was carefully added via the top of the chamber

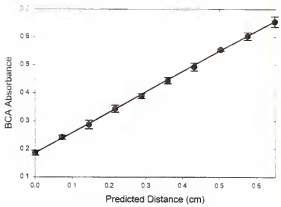

Fig. 3.2. Assessment of linear gradient-forming protocol. A BCA protein gradient was formed using the gradient forming protocol. BCA-collagen gel was then cut into equal slices along the gradient region. The mean absorbance of three measurements/slice according to spectrophotometer anaylsis is reported. Plot of predicted distance (*based on equally sized slices and a total gradient length of ≈.65cm in length*) versus BCA standardize curve (solid line) and mean absorbance of BCA gradient slices (•). Error bars represent ± standard error.

and chamber was re-sealed as described above. Finally, gradient chamber was placed onto the stage microscope and the cell-tracking experiments began.

Quantification of adhesive gradients was accomplished using a bicinchoninic acid (BCA) protein assay reagent (Sigma Chemical Company, St. Louis, MO) Bicinchoninic acid forms a blue copper complex in the presence of proteins (collagen in this case). To test the gradient characteristics, a collagen gels containing gradients of BCA protein determination reagent (as opposed to RGD gradients) were formed using the exact automated gradient forming procedure as described above except here solution *B* was collagen stained with BCA protein determination reagent (solution *A* was unmodified collagen, 0% BCA). In solution *B*, the BCA protein reagent replaced the DMEM. The gradient forming procedure created a gradient of increasing BCA protein reagent concentration that could be quantified spectrophotemetrically. To determine the distribution of BCA in the chamber, the BCA gradient collagen gel was cut into ten slices along the length of the BCA gradient region. Individual segments of the slices were then dissolved under gentle sonication. Using spectrophotometer analysis (BCA λ=562nm), the absorbance of each slice was measured and compared to a set of BCA collagen standards to test the linearity of the gradient (Fig. 3.2). The apparent linear

relationship between increasing BCA absorbance (higher absorbance is result of increased concentration of BCA) as a function of increasing distance along the gradient region implies that the gradient forming protocol was reproducible and accurately described as linear gradient, at least to the detection level of the spectrophotometer.

Three-Dimensional Cell Tracking

For direct observation of cell migration, the positions of multiple individual cells were tracked in real time using the automated video microscopy system along the length of the linear gradient. The tracking methodology has previously been described (Burgess et al., 2000). Briefly, a computer program was written to coordinate the movements of the motorized stage and focus control motors, and processed digital images in order to track the positions of individual cells in an automated fashion. To accomplish this, the initial three-dimensional positions of the individual cells to be tracked were found manually using the motorized stage and recorded to an output file. After locating the initial positions of all cells to be tracked, the computer algorithm recursively returned to the previous position of cells and updated the three-dimensional position of each cell. A series of image processing techniques was completed to locate and update the cell's center of mass. Following this, the microscope stage was re-centered such that the cell's center of mass was in the center of the projected image and the updated position of the cell was recorded to an output file. This was recursively executed for all cells and all time points. For each cell tracking experiment a total of 80 cells were tracked along the length of the adhesive gradient. The time interval between measurements for each cell was approximately 35 minutes. Multiple cell tracking experiments for each adhesive gradient steepness were completed.

Statistical Analysis of Cell Paths

To determine if cells exhibited biased migration, a number of statistics were generated from the cell paths to quantify the time-average and population-average migration behavior for each experiment as a function of increasing adhesive gradient steepness. From theory for isotropic migration (random motility), it is expected that the time-averaged x, y, and z displacements are equal to zero. However, in the presence of a gradient stimulus the same may not hold true. Accordingly, to quantify if a significant bias in the migration resulted due to the adhesive gradient *(haptotaxis migration)*, the average drift velocity, V_D, was calculated according to (Lauffenburger, 1983):

$$V_D = \phi \cdot S \tag{3.1}$$

where ϕ represents average orientation bias of cells and S is cell speed. V_D is a measure of the rate of cell redistribution towards *or* away from increasing adhesive gradient steepness.

For isotropic migration $\langle \cos\theta \rangle$ and $\langle \sin\theta \rangle$ are expected to equal to zero. However, in the presence of a cell stimulus gradient, a preferred orientation alignment, ϕ, can result. Thus the time-averaged orientation, ϕ, can be defined (Dedhar et al., 1987):

$$\phi = \sum_{j=1}^{M} \frac{1}{N_0 - 1} \sum_{i=1}^{N_0} \frac{\Delta x_{i,j}}{\sqrt{\Delta x_{i,j}^2 + \Delta y_{i,j}^2}} \tag{3.2}$$

where Δx and Δy represent distance cell moved in one time increment in the x and y axis direction respectively, N_0 represents the number of time points and M is the total number of cells tracked. The maximum degree of orientation is 1, which represents the case in which cells are aligning perfectly towards increasing gradient direction. Similarly, a value of ϕ = -1 represents cells perfectly aligning in the direction of decreasing gradient concentration. A value of ϕ = 0 represents no significant alignment of cells.

Average cell speed, S, can be calculated by the following (Bultmann and Gruler, 1983):

$$S = \sum_{j=1}^{M} \sum_{i=1}^{N_j} \frac{\sqrt{\Delta x_{i,j}^2 + \Delta y_{i,j}^2}}{\Delta t} \tag{3.3}$$

where dt is the time increment between measured cell displacements. The haptotaxis coefficient, χ, which describes the intrinsic ability of the cells to bias their direction of movement in the direction of increasing gradient stimulus can also be calculated. It proportional to the V_D and gradient concentration (Lauffenburger, 1983):

$$V_D = \chi \cdot \frac{dC}{dx} \tag{3.4}$$

where dC/dx represents the change in RGD-concentration per unit length.

In addition to a potential significant drift velocity and/or orientation alignment due to a gradient stimulus, enhanced cell speed parallel to the gradient stimulus (*orthotaxis*) has been reported. It should be noted that from theory for isotropic migration, it is expected that the mean-root square velocity components (V_x, V_y, V_z) are equal (Alt, 1990). To determine if the presence of the adhesive gradient affected the velocity of the cells, the time-averaged mean-root square velocity, $< \sqrt{V_x^2} >, < \sqrt{V_y^2} >$, was calculated:

$$< \sqrt{V_i^2} > = < \sqrt{(\frac{\Delta i}{\Delta t})^2} > \tag{3.5}$$

where i represents x, y or z-axis.

Birefringence Measurements

A series of birefringence experiments were conducted to determine if there was a biased alignment of the collagen fibrils within the haptotaxis chamber. To do this, the key components included polarized light from a 17mW He-Ne laser (Melles Griot; Irvine, CA) that was radiated through the collagen sample. The collagen sample was securely mounted on a rotating stage

(NRC; Fountain Valley, CA). The transmitted light through the collagen sample was measured using a photodiode and a calibrated digital voltmeter (Keithley; Cleveland, OH). Care was taken to insure light source and collagen sample was properly aligned with respect to each other prior to beginning the experiments. Intensity measurements were taken as a function of rotated angle by precisely rotating the calibrated stage at five-degree incremented rotations between 0° and 180°. For each data point, the mean and variance based 100 intensity measurements (sample rate =10Hz) with both the laser on and off. The variation in the mean intensity measurements between laser on/off modes represents the mean transmitted radiated light, β. Control experiments were completed by pipetting a collagen solution into similar chamber and heat polymerized with the chamber oriented in a horizontal position. For a more direct comparison of results between experiments, differences (such as cell culture media pH, unclean/scratched glass chamber surfaces, etc), which could potentially affect the value of β, were minimized by normalizing the mean intensity for each run by the intensity value at 0° for more direct comparison between vertically pumped gels and horizontally gelled collagen solutions. Data experiments were repeated at multiple points along the chamber length to check for uniformity of the gel.

Results

The focus of this study was to characterize cell migration in response to three-dimensional physiologically relevant spatial adhesive gradients as a function of increasing adhesive gradient steepnesses. Recently Myles et al. (2000), demonstrated that the adhesive properties of Type I collagen could be enhanced for K1735 murine melanoma cells through the grafting of a well-documented adhesive RGD peptide sequence (glycine-arginine-glycine-aspartic acid serine-proline-cysteine, hereafter referred to as RGD) (Ruoslahti and Pierschbacher, 1987; Ruoslahti and Pierschbacher, 1986). Properly controlling the cross-linking reactions (e.g.,

reaction times, amount of cross-linker and RGD peptides) the degree of conjugation could be precisely controlled (Myles et al., 2000), thus providing a method to control cell-substratum adhesiveness relative to collagen.

For this study, three degrees of RGD conjugation to collagen was used: 8, 12, and 22 peptides grafted per collagen monomer (hereafter referred to as 8x, 12x and 22x). Since the RGD-grafted collagen provided increased adhesiveness compared to unmodified collagen, it was possible to create an adhesive gradient by mixing a collagen solution (*no grafted RGD peptides*) and an RGD-grafted collagen solution to form a linear three-dimensional adhesive gradient. A protocol was devised to precisely control gradient steepness using a computer programmed peristaltic pump equipped with a gradient forming function in which a collagen solution and an RGD-grafted collagen solution (either 8, 12, or 22 RGD peptides grafted to collagen) were mixed proportionally to form a linear RGD-concentration gradient. Based on this, the theoretical gradient steepnesses used in these experiments were 88, 132 and 241 nmol RGD peptide/ml*cm. In addition as an experimental control, a nonadhesive peptide composed of glycine-arginine-glycine-glutamic acid serine-proline-cysteine (GRGESPC, hereafter referred to as RGE) was also grafted to collagen (Dedhar et al., 1987; Pytela et al., 1986; Pierschbacher and Ruoslahti, 1984). This peptide was both similar in size and structure with the deletion of aspartic acid and the insertion of glutamic acid. The grafting of RGE to collagen for a degree of conjugation equal 22 peptides grafted per collagen monomer has previously been reported to not significantly enhance the adhesiveness of collagen (Myles et al., 2000). Using the RGE-grafted collagen with 20 RGE peptides grafted to collagen (hereafter referred to as 20x), RGE-concentration gradients were also formed with theoretical gradient steepness equal 219 nmol RGE peptide/ml*cm. Utilizing RGD-gradient assays (adhesive gradient) and RGE gradients (nonadhesive gradients) provided a method to systematically evaluate migration of cells in response to adhesive gradients and to determine if the response could be attributed to the adhesive gradient or the grafting process. In

addition to the RGE-concentration gradients as a control, replacing the RGD/RGE-grafted collagen solution with unmodified collagen solution in which both solutions were composed of unmodified collagen created a simulated collagen gradient. Using the same gradient-forming protocol, this provided a means to investigate if experimental artifacts were arising due specifically to the gradient forming protocol.

Due to the nature of the pumping action of peristaltic pumps, cell viability was a concern. To determine if this step in the gradient forming protocol was harmful to the viability of the ells, standard typan blue assay tests were used. It was determined a pre-wash using PBS, followed by a collagen pre-coating of the inner silastic tube lumen applied at a slow pumping rate, the pumping protocol was not significantly harmful to the cells. Based on the long-term survival of cells during over the course of tracking experiments additionally suggested the cells were not observably impaired.

Utilizing the RGD/RGE concentration gradient migration assays with cells interdispersed within the cell matrix, multiple cell migration experiments were completed as a function of gradient steepness. From the cell tracking experiments the time-incremented cell displacements was calculated to determine the time-averaged cell population average response to the RGD/RGE and simulated gradients. Based on the time-incremented average cell displacements, it was observed that cells did exhibit a small but significant bias towards increasing RGD-gradient concentration indicated by the average drift velocity, V_D (Fig. 3.3).

For the RGE-concentration gradient control experiments, no significant V_D was observed, suggesting the biased migration was attributable to the adhesive gradient. Further, for the simulated collagen gradients no significant bias was observed, suggesting the gradient forming protocol was *not* the cause of biased migration. Together these data strongly suggest the V_D observed for the RGD-gradients was due specifically to the adhesive properties of the gradient

rather than a secondary effect due to artifacts of the grafting procedure or the gradient-forming protocol.

Cells can potentially modify their paths of normal random motility in a number of ways to elicit a biased migration response. To quantify how these cells modified their paths resulting in a significant biased migration described by V_D, a detailed analysis of individual time-incremented cell displacements is required. One potential mode of biased migration is through a preferential orientation of cells in the gradient direction. The result is, though the cell may move at a constant speed, it has a higher frequency of orienting in one direction resulting in a net displacement towards the direction of preferential alignment. Investigating if this might have contributed to or was responsible for the observed biased migration, the average degree of orientation, ϕ, was calculated according to Eqn. 3.2. Values of ϕ=1 or -1 represent a cell migrating perfectly aligned towards or away from increasing gradient concentration respectively. Based on Figure 3.4, a small but significant biased alignment of cells towards increasing RGD-gradient concentration was observed. Although ϕ was found to be small in magnitude, these results suggest that biased orientation contributed to haptotaxis migration. This type of biased migration has been previously termed tropotaxis (biased orientation of cells) (Dickinson and Tranquillo, 1993b). Related to this, the haptotaxis coefficient, χ (describes the intrinsic ability of the cells to bias their direction of movement in the direction of increasing gradient stimulus, ml/mol peptide•s) (Tranquillo et al., 1988), was also calculated based on Eqn. 3.4. Values of χ for RGD-adhesive gradients were found to be

significant; however, no strong dependency on adhesive gradient steepness was observed (Fig. 3.4).

Another potential method cells can modify their paths to elicit a biased migration response is through a dependence of cell speed on the alignment of cells (orthotaxis). Simply, the idea is that cell speed can be potentially increased/decreased depending on the orientation of the

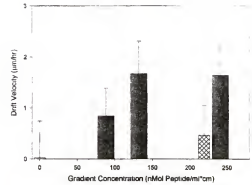

Fig. 3.3. Multiple individual cell paths were analyzed as a function of gradient steepness. Cells exposed to RGD-concentration gradients were found to exhibit a significant drift velocity, VD, in the direction of increasing concentration. No significant VD was observed for RGE and simulated gradients. Error bars represent ± standard error.

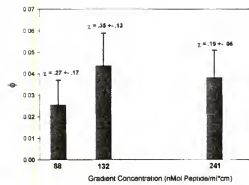

Fig. 3.4. Orientation alignment of cells was analyzed with respect to gradient direction. Cells exposed to RGD concentration gradients were found a have a small significant biased alignment in the direction of increasing gradient steepness indicated by ϕ. Haptotaxis coefficient, χ (ml/mol peptide•s), which describes the intrinsic ability of cells to bias their direction of movement towards increasing gradient stimulus is given for each case. Error bars represent ± standard error.

cell with respect to the anisotropic environment. Investigating this, we found cell speed, *S*, was found biphasically dependent on orientation alignment with maximal cell speeds observed when aligned along the major axis of the gradient (increasing gradient concentration is 0°) (Fig. 3.5). A bi-directional dependence of cell speed as illustrated in Fig 3.5 suggests cells moved at faster rates both towards and away from increasing gradient concentrations. Increased *S* for both directions along the gradient is not expected for these types of experiments and is not consistent with an orthotaxis effect. Instead, these results are more consistent with previous results reported for cells migrating on oriented substratum such as oriented collagen fibrils (contact guidance) (Guido and Tranquillo, 1993; Dickinson et al., 1994; Tranquillo et al., 1996).

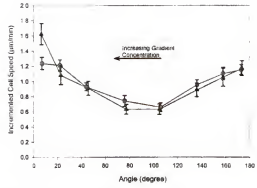

Fig. 3.5. Relationship describes the dependence of average incremented cell speed (μm/min) as a function of orientation of the cell with respect to increasing gradient stimulus for 88 (▲) and 241 (•) nmol RGD-concentration gradient. Error bars represent ± standard error.

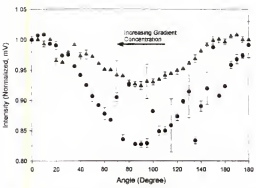

Fig. 3.6. Birefringence intensity (mV) measurement results as a function of rotation angle (degrees) for vertically pumped (•) and horizontal polymerized (▲) collagen gels. Long axis of chamber in the 0° – 180° plane. Intensity values are normalized by intensity value at 0° for each case. Biphasic dependence of transmitted light as a function of rotated angle suggests partial collagen fibril alignment along the length of the gradient chamber (long axis). Error bars represent ± standard deviation.

Finally, another potential mode a cell can modify their cell displacement is through a dependence of cell speed on the magnitude of stimulus (adhesion in this case). This effect is known as orthokinesis. Based on our data, however, *S* was not found to be significantly dependent in any consistent fashion on the magnitude of the gradient, suggesting orthokinesis did not contribute to biased migration.

The results presented in Fig 3.4 are suggestive of a preferential alignment of collagen fibrils along the major axis of the chamber. To determine if this was the case, a series of birefringence experiments were performed. As an experimental control, collagen gels were also prepared by pipetting solutions into the gradient chamber and heat polymerized in a horizontal position (instead of vertical position as described in protocol) to eliminate possible effects of pumping and/or gravity. To test for birefringence, polarized light from He-Ne laser mounted on a

rotating stage was radiated through the each chamber. Mean transmitted light, β, was collected and measured using a photodiode and high-resolution digital voltmeter as a function of rotated angle. A significant alignment of the collagen fibrils in one direction would effectively acts as a light polarizer on transmitted light through the sample. Light passed through two light polarizers that are rotated 90° apart results in complete blocking of the transmitted light. Likewise, for two polarizers perfectly aligned 100% of the polarized light is transmitted through the polarizers. Taking advantage of the physics of polarized light, here in our experimental design the first polarizer is attached to the laser and the second potential polarizer is the collagen fibrils. If the fibrils are aligned then they will act collectively as a light polarizer. Using this experimental design, it was determined that β was in fact angle-dependent for the case of vertically pumped chambers (Fig. 3.6). Furthermore, a minimum β was observed at the rotated angle of 90°, which represents the point at which the polarized light and the major axis of the gradient chamber are perpendicular to each other. These results suggest that not only is there is at least some partial alignment of the collagen fibrils but also the collagen fibrils were preferentially oriented along the major axis of the chamber (in the direction of the gradient).

The average change in intensity of light measured comparing the parallel (0°, 180°) and orthogonal directions (90°) with respect to the orientation of the gradient chamber represented roughly a 15% difference, suggesting that the orientation of the fibrils were weakly oriented along the major axis of the chamber. Interestingly and unexpected, for the control sample, birefringence measurements also suggested a small angle-dependence for transmitted light resulting in an average difference between the parallel and orthogonal directions being 6%. This partial alignment for the control sample could be attributed to geometric effects of the chamber itself or possible measurement errors. In summary, these results strongly suggest the biphasic dependence for cell speed as reported in Figure 3.5 is, at least in part, due to the partial alignment of collagen fibrils.

Discussion

The correlation between active cellular movements and cell-stimuli has been widely reported for a number of cell types. In the case of normal functioning tissue cells, a tissue cell migrates by crawling over substrata. Cellular crawling is mediated by the cell attaching to the substratum and pulling itself forward. If insufficient traction is provided to cell, the cell is less effective in its ability to pull itself forward. Likewise, if too much traction is provided, intracellular propulsion forces are not able to effectively break attachments between the cell and substrata. With this physical picture in mind, it has been proposed that by properly modulating the adhesive properties of the cell-substratum one can engineer directed cellular movements. A number of studies have been completed investigating how cells respond to adhesion gradients. However, to date these experiments have been limited to a two-dimensional geometry and indirect observations of cell migration. This aim of this study was to investigate by direct observation how cells respond to an adhesive gradient in a three-dimensional cell matrix.

Utilizing the coupling procedure developed by Myles et al. (2000) provided a method to systematically amplify the adhesive properties of collagen. Cell adhesion for murine melanoma cells (used in this study) has been shown to linearly increase as the number of grafted adhesive peptides to collagen increased (Myles et al., 2000). Exploiting this coupling procedure to control adhesiveness of collagen, a protocol was developed in which two collagen solutions of differing adhesiveness was mixed by linearly changing the volumetric proportions of each solution. By slowly pumping these mixed solutions into the gradient chamber and then heat polymerizing the collagen solution, presumably an adhesive gradient was created. Since directly testing the adhesive properties within a three-dimensional matrix would be both technically very complex and difficult to interpret the results, no direct tests of the adhesive properties of the three-dimensional gradients were completed. Related to this, however, Myles et al. (2000)

demonstrated that for two-dimensional cell adhesion, adhesion increased as the volumetric ratio of RGD-grafted collagen/unmodified increased. For the validation of a linear RGD-concentration gradient, the gradient protocol was tested using the BCA test reagent and spectrophotometer, and, at least to the precision of the spectrophotometer, the data suggests a linear RGD-concentration was created. Together these data strongly suggest the protocol developed was effective in creating a three-dimensional adhesive gradient.

Was the steepness of the adhesion gradient sufficiently steep for the cells to detect and perceive the direction of gradient? The mechanism(s) by which a cell detects a gradient is not well understood, but for many cells it has been reported they respond not to an absolute change in concentration (dC/dx) but rather to the proportional change in concentration ($dC/C/dx$) (Tranquillo, 1990; Brandley and Schnaar, 1989; Zigmond, 1981). Consistent with this idea, the gradient steepness can be characterized by $\varepsilon = 100(\Delta/C)$ (Tranquillo, 1990), which represents the fractional percent molar change of adhesive peptides across the effective cell diameter (taken as 25 μm), where Δ is the concentration difference and C is the concentration at the cell midpoint. Based on gradients used in these experiments, $\varepsilon \approx 10$ -.4% from the lower region of the gradient to the upper region (proportional change for a linear gradient is not constant). Turning to previous studies reported in literature, others have reported minimum gradient steepnesses for cell perception. Zigmond et al. (1981) reported for neutrophil chemotaxis using gradients of N-formylated chemotactic peptides a significant orientation of cells towards higher chemoattractant concentration for fixed gradient steepness of $\varepsilon \approx 1\%$. Nuccitelli and Erickson (Nuccitelli and Erickson, 1983) have also reported for migratory embryonic quail somitic fibroblasts sensitivity to electric field gradients (*galvanotaxis*) for a value of $\varepsilon \approx 10\%$. More recently Brandley and Schnaar (Brandley and Schnaar, 1989) reported significant redistribution for two-dimensional migration of B10 murine melanoma cells on linear immobilized adhesive gradients with highest accumulation of cells was equal to a value $\varepsilon \approx 3.5\%$. Together these results suggest the range of

adhesive gradient steepness used in our experiments was sufficient steep for the cells to perceive the gradients.

Interestingly enough, the incremented cell speed, *S*, for the 8x gradient was the highest for the upper region of the gradient (50-100% 8x) corresponding to $\varepsilon \approx .5$ % and an absolute gradient steepness of approximately 43nmol RGD/ml. Comparing this to the 22x linear gradient, the higher *S* was observed in the instead in the first half (0-50% 22x) of the gradient region with an average value of $\varepsilon \approx 3.5$ % and absolute gradient steepness of approximately 39nmol RGD/ml respectively. Comparisons of these results suggest that, at least for cells used in this study over the duration of the tracking experiments, the cells do not adapt to the level of adhesiveness (if this were true one would expect both $\varepsilon_{8x} \approx \varepsilon_{22x}$) but instead to respond to the absolute concentration of the adhesion peptide (dC/dx). The suggestion of the cells responding to the absolute concentration gradient rather than proportional to the change in concentration is consistent with previous reported results by Burgess et al. (2000) and others (Huttenlocher et al., 1995; Palecek et al., 1997; DiMilla et al., 1993) demonstrating a biphasic dependence of cell motility on the adhesiveness of the substratum. It should be pointed out, however, this is not to imply cells would not or do not respond to a proportional changes in the gradient concentration, simply over the time scale of experimentation in this study, our data does not support this conclusion.

Collagen fibrils were unexpectedly found to be partially aligned. These findings (Fig 3.6) casts light on possibly explaining why the cell speed was increased for directions along the major axis of the chamber and decreased when the perpendicular to the major axis. Previously, the migration of cells along aligned collagen fibrils has been shown to result a bi-directional biased migration along the axis of alignment for reasons hypothesized to be related to decreased cellular drag resistance in that direction (Dickinson et al., 1994). In these current experiments, the major axis of alignment is also parallel to the adhesive gradient. Considering what might potentially cause an alignment of the fibrils, previously it has been shown that gravity can cause

some partial alignment of collagen and is likely the primary contributing factor for the observed fibril alignment (personal communication Dr. Robert Tranquillo's research group). In essence, an alignment of collagen fibrils could result in cells migrating at faster rates when aligned parallel to the fibrils and at slower rates when perpendicular to fibril alignment. Such an effect is consistent with our data (Fig. 3.5). In contrast, however, fibril alignment would not be expected to result in a unidirectional biased migration indicated by V_D along (Fig 3.3) and therefore we feel an aligning of the collagen fibrils cannot explain these results. Additionally, our point is further emphasized by the fact that of no significant V_d was associated with either the nonadhesive gradient or the collagen simulated gradient both of which were created following the same exact protocol as the RGD-gradients. Instead, we feel the partial alignment of collagen fibrils, might result in the cells having a slightly higher probability of being oriented along the gradient axis and as a result, the partial alignment could potentially act as aligning cue orienting the cell parallel to along the gradient axis.

A net directional drift towards increasing gradient concentration was observed for all adhesive gradients tested while no significant redistribution was found for either the nonadhesive gradient and simulated gradient controls, strongly suggesting the cells were responding to the adhesiveness of the gradient rather than an artifact either from the grafting procedure or in the gradient-forming protocol such as collagen fibril alignment. Though the cells did exhibit significant V_d for the adhesive gradients, the rate of redistribution was found to be an order of magnitude lower than reported for cells interspersed in uniform collagen gels (Burgess et al., 2000) suggesting that the cells ability to perceive the direction of increasing gradient concentration was less efficient. The explanation of this could be related to mechanism of multiple competing pseudopods in which cell extend and retract by pulling on the pseudopods simultaneously. (DiMilla et al., 1991; Dickinson and Tranquillo, 1993b). Pseudopods supported by higher adhesion to the substrata eventually dominate and pulls the cell forward in the direction

of the dominant pseudopod. If, as in the adhesion gradient, the adhesiveness is high, the cell's ability to break attachments is diminished and the perception of the direction of the gradient is diminished. The net result is a decreased cell speed and lower V_d. For the case of adhesion of melanoma cells to collagen, it has previously demonstrated adhesion indeed is high potentially creating a scenario in which it would be difficult for the cells to accurately perceive the concentration gradient (Burgess et al., 1999). This may provide an explanation as to the reason no significant difference in the average V_d for the range of adhesive gradients investigated was found (Fig. 3.2). The mechanism of competing pseudopods for the case of high baseline adhesion would also predict little, if any, difference in V_d.

In conclusion, this is the first study, to our knowledge, for the direct observation of three-dimensional haptotaxis cell migration. Our results support the hypothesis that adhesive gradients can directly affect the cellular movements and can result in a significant redistribution of cells towards increasing gradient concentration. In contrast to previous reports for chemotaxis gradients, our results do not lend support the hypothesis that cells respond to proportional changes in the gradient steepness. Instead our results suggest for the case of adhesive gradients, a response to absolute concentration changes in the gradient steepness (dC/dx). Based on birefringence tests, the data suggest partial alignment of collagen fibrils in the adhesive gradient chamber. This could act as orientation cue, tending to orient the cells parallel to the direction of the gradient and would be expected to enhance the measured V_d had there been no alignment. Finally, although only small rates of redistribution towards increasing gradient concentrations were found, our results do provide an optimistic outlook for the use of adhesive gradients in therapeutic applications.

CHAPTER 4
THE "ACTIN FILAMENT ZIPPER": A BIOPHYSICAL MODEL FOR ACTIN-BASED MECHANOCHEMICAL PROPULSION

Introduction

Cellular movements such as extension of membrane filopodia and lamellipodia, and the intracellular propulsion of *Listeria monocytogenes* are largely determined by the actin-containing cytoskeleton (Zaner, 1995). Functionally, actin filaments serve as intracellular scaffolds that promote both locomotion and force transduction. Actin filaments are now recognized as semiflexible biopolymers that undergo bending undulations due to thermal fluctuations (Kas et al., 1996). The mechanical rigidity of actin filaments can be characterized by a theoretical bending modulus $B=\lambda p k_b T$ (Doi and Edwards, 1986) where λ_p is known as the persistence length, a quantitative measure of filament stiffness. For actin filaments, values for λ_p have been experimentally observed to range from .5μm - 15μm (Isambert et al., 1995; Götter et al., 1996; Kas et al., 1996). In addition to being characterized as semiflexible rods, actin filaments also interact directly with a number of actin-binding proteins to form highly ordered complex structures. One class of actin-binding proteins is known as actin bundling proteins (ABP) (Vandekerckhove and Vancompernolle, 1992; Ayscough, 1998). These proteins can interact laterally with actin to form filament-filament crosslinked actin bundles. Experimentally, it has been demonstrated that a randomly oriented actin network, when exposed an ABP such as α-actinin, can undergo dynamic rearrangement of the actin filaments to form parallel actin bundles (Wachsstock et al., 1993; Bartles, 2000).

Physically, the cross-bridging of two nonparallel filaments can result in a local interlocking of the actin filaments to a collinear configuration creating a flexural bend in the two nonparallel actin filaments subsequently generating in a pushing force. We call this process the "Actin Filament Zipper" (AFZ) because it is the transduction of chemical energy into a mechanical energy through the cross-bridging of actin bundling proteins to actin resulting in a zippering of actin filaments. Three experimental studies in particular support the AFZ mechanism as a potentially an important mechanism in cellular movements. First, studying the intracellular motility *Listeria monocytogenes*, Dold et al. (1994) discovered microinjecting the 53kDa proteolytic fragment of the ABP α-actinin resulted in inhibiting the motility of Listeria, strongly suggesting that dynamic and regulated interactions of α-actinin is a critical component in the intracellular propulsion of Listeria. Similar results of disrupted cytoskeleton and disassembly of stress fibers have also been reported for fibroblasts and fibroma cells for microinjections involving the proteolytic fragments of α-actinin (Pavalko and Burridge, 1991). Second, high-resolution electron microscopy observations of the actin rocket-like tail of the *Listeria monocytogenes* by Sechi

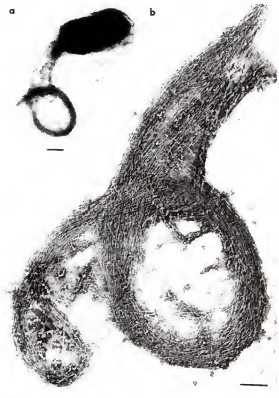

Fig. 4.1. Isolated macrophage pseudopodium (*a*), at higher magnification in (*b*), that shows a prominent axial array of long actin filaments. Bars: (*a*) 0.5 μm; (*b*) 0.2 μm. Taken from (Sechi et al., 1997).

Table 4-1 Parameter Values

Notation	Definition	Value	Reference
k_f	Forward binding rate, Eqn. (4.1)	6.8×10^6 $(Ms)^{-1}$	(Goldmann and Isenberg, 1993; Wachestock et al., 1993)
k_r	Reverse binding rate, Eqn. (4.1)	.4 s^{-1}	(Goldmann and Isenberg, 1993; Wachsstock et al., 1993)
R_T	Mean density free α-actinin binding sites/length actin filament	.025 nm^{-1}	(Meyer and Aebi, 1990)
λ_p	Persistent length of actin	1-15μm	(Kas et al., 1996; Götter et al., 1996; Isambert et al., 1995)
l	Length of α-actinin molecule	50nm	(Podlubnaya et al., 1975; Jockusch et al., 1995)

et al. (1997), show the physical re-arrangement of actin filaments in the actin network from randomly oriented near the bacterial cell wall to increasingly collinearly aligned actin bundles with increasing distances from the bacterium (Fig. 4.1). Immunofluorescence microscopy experiments by Sechi et al. (1997) revealed this dynamic alignment of the actin filaments was due to crosslinking by α-actinin. Third, actin bundling activity of villin has been reported to induce the formation of microvillar structures typical in brush borders in transfected fibroblasts (Friederich et al., 1989).

It is increasingly clear a number of cellular movements do not require molecular motors (e.g., myosin). To explain these types of movements, it is frequently asserted actin filaments and actin-associated proteins are responsible for the force production. How the cell harnesses these cytoskeleton components to exert a propulsive force, however, is still not completely understood. One proposed explanation, using thermodynamic arguments, points to the free energy, ΔG, released through the addition of a single monomer to an actin filament estimated to be on the order of $-10k_bT$ (Condeelis, 1993), however, no underlying physical mechanism is described as to how this energy is translated into a pushing force. Peskin et al.

Table 4-2 Symbol Notations

Symbol	Definition
k_BT	Thermal energy, 4.1pN/nm
γ_s	Spring constant, .12pN/nm
γ_b	Chemical bond spring constant, (pN/nm)
γ_{rel}	Linear repulsion decay, 1 pN/nm
r_{min}	Relaxed spacing between beads, 100nm
r	Vector position along the filament
d	Separation distance between two actin filament, (nm)
d_0	Relaxed bond separation distance (nm)
z, z_0	Dimensionless d, d_0
L	Length of actin filament (nm)
E_B	Bending energy of actin filament, (pN*nm)
ΔG_{rxn}	Gibbs free energy, (pN/nm)
μ	Chemical potential, (nm^{-1})
μ_0	Standard state chemical potential, (nm^{-1})
B	Bending modulus ($pN*nm^2$)
s	Unit vector along length of filament
u	Unit vector tangent at contour length s
C_1	Density of singly-bound α-actinin molecules/length of actin filament, (nm^{-1})
C_2	Density of doubly-bound α-actinin molecules/length of actin filament, (nm^{-1})
C_3	Density of singly-bound CAB molecules/length of actin filament, (nm^{-1})
U_1, U_2, U_3	Dimensionless C_1, C_2, C_3 respectively
R	Density of free α-actinin binding sites/length (nm^{-1})
A	Alpha-actinin concentration, (M)
I	CAB species concentration (M)
a	Dimensionless α-actinin concentration, A/K_D
g	Dimensionless CAB species concentration, I/K_D
ρ	Dimensionless crosslinking affinity
α	Dimensionless disassociation rate, Eqn. 1
ω	Relative dissociation rates of Eqn 10 and 4 respectively
k_{10}	Unstressed cross-bridging association rate, $(Ms)^{-1}$
k_1	Association binding rate, Eqn 2 (#/nm*s)
k_1	Association rate of Eqn. 10; assumed equal to k_1 $(Ms)^{-1}$
k_d	Disassociation rate of Eqn 10; assumed equal to k_1 (s^{-1})
K_D	Equilibrium disassociation constant = k_d/k_1 (M)
k_{r2}	Association crosslinking binding rate $(Ms)^{-1}$
k_{r0}	Unstressed cross-bridging association rate; assumed $10k_1$
t	Time, (s)
τ_1	Dimensionless time constant for Thermodynamic model, $k_1 t$
τ_2	Dimensionless time constant for BD model, Dt/r_{min}^2
D	Diffusion coefficient of bead, K_BT/δ (nm^2/s)
δ	Bead drag coefficient, $6\pi\mu_{eff}r$
μ_{eff}	Effective viscosity, .1g/cm*s
r	Bead radius, 50nm
dW	Increment in Weiner process
F_i	Deterministic forces acting on i^{th} bead, (pN)
$F_{B,i}$	Bending force on the i^{th} bead, (pN)
$F_{s,i}$	Spring force on the i^{th} bead, (pN)
$F_{chem,i}$	Chemical interaction force on the i^{th} bead, (pN)
$F_{rep(i)}$	Repulsive force on the i^{th} bead, (pN)

(1993) formulated a mathematical model described as the Brownian ratchet that accounts for the one-dimensional diffusion of a particle with infinitely stiff polymerizing actin filaments buttressed at one end of the particle. Peskin et al. proposed that Brownian movements away from rigid actin filaments could potentially create a temporal spatial gap between the particle and the filaments sufficient to permit the insertion of a new monomer, thus resulting in a directed axial "push" of the particle in the direction of the growing actin filaments. Based on the Brownian ratchet model, Mogilner and Oster (1996) proposed a modified model of Peskin et al.'s earlier model that includes the elasticity of the actin filaments and the Brownian

fluctuations of the individual actin filaments (*instead of the particle*). Using this model, they predicted a pushing force ~.06pN/filament. This model provides many insights into how actin filaments actin polymerization can be utilized to exert a surface pressure. However, the Brownian ratchet model does not directly model the role of actin-binding proteins in both providing rigidity to actin filaments through cross-bridging interactions and in their apparent ability to organize and orient actin filaments into a number of diverse structures. Recently, Taunton et al. (2000) reconstituted an actin-based motility assay *in vitro* using lysomes and endosomes. Interestingly, the morphology of the deformed vesicles suggests actin filaments exert pressure on the sides rather than the rear, consistent with a "squeezing" mechanism rather than a pushing mechanism. These observations suggest that the forces exerted by the actin filaments may fully captured by a Brownian ratchet mechanism and may involve an additionally and/or alternative biophysical mechanism.

In this chapter we present two mathematical mechanochemical models describing the AFZ mechanism. In the first model, a thermodynamic model is presented which accounts for binding kinetics between actin and ABPs, bond stiffness, and the mechanical properties of actin filaments. In the second modeling approach, a Brownian dynamics (BD) model is presented which incorporates the BD movements of the individual actin filaments in addition to accounting for the reaction kinetics and the mechanical properties of the actin filaments. Using the intracellular actin-based motility of *Listeria monocytogenes* as our motility model, we will focus among the ABPs specifically on α-actinin because it has been reported in high concentrations decorated along the length of actin filaments in the actin rocket-like tail (Dabiri et al., 1990; Sechi et al., 1997) and we feel it plays an important role in the intracellular propulsion of Listeria. To characterize the AFZ mechanism we will derive a relationship describing force versus both α-actinin and a competitive actin-binding inhibitor

concentration as a function of mechanical stiffness of actin filaments, bond stiffness, and binding affinities.

Mechanochemical Model for the Actin-Zipper Mechanism

Adopting these described mechanical and chemical properties of actin and ABPs, our model is herein built upon the assumption that dynamic and regulated rearrangements of actin filaments due to cross-bridging interactions between actin and ABPs can consequent in flexural bending of the individual actin filaments during the re-alignment process. Physically, the bundling of two nonparallel filaments results in a local interlocking of the actin filaments to a parallel configuration. This parallel

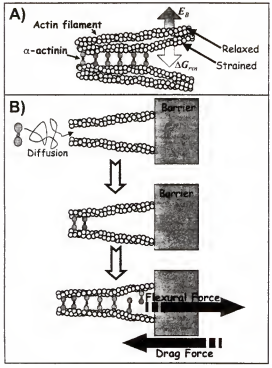

Fig 4.2. Schematic of the "Actin Filament Zipper" hypothesis. A) Accounting for the interaction of between actinin and actin filaments the chemical energy energetically driving the bundling of actin filaments, described by ΔG_{rxn}, is in equilibrium with the mechanical energy associated with bending an actin filament, E_B. B) Bundling of actin filaments is described in three steps: 1) diffusion to the actinin to actin filaments, 2) binding of actinin to actin forming cross-bridging bonds between actin filaments 3) continued bundling via actinin along the length of the actin filaments. Mechanical pushing forces are generated through the flexural bending of actin filaments resulting from the dynamic bundling and aligning actin filaments to a collinear configuration. The cytoplasmic drag force exerted on the barrier balances mechanical pushing forces.

aligning of the actin filaments due to cross-bridging bonds can cause a flexural bend in the actin filament(s) potentially resulting in a pushing force on the load. Hence, the chemical

energy due specifically to the cross-bridging interaction of ABPs energetically drives the AFZ mechanism imparting pushing force. The chemical energy of the system is balanced by the mechanical energy associated with bending an actin filament (Fig. 4.2A). In essence, it is the transduction of chemical energy into mechanical energy due to the bundling of actin filaments that is hypothesized capable of generating significant cellular forces.

Two modeling approaches are used to investigate the AFZ mechanism. In the first approach, a thermodynamic model describing the AFZ mechanism is formulated. The reaction kinetics describing the bundling of two actin filaments is modeled as a two-step reversible reaction. The highlight of this model will be to derive a relationship for propulsive force versus α-actinin concentration as a function of mechanical stiffness of the actin filament, cross-bridging bond stiffness and binding affinities. The second model will additionally account for the thermal fluctuations of individual actin filaments. To model these fluctuations, the classical Hookean bead-spring chain model developed by Rouse (Ottinger, 1996) is used. The goal of the Brownian dynamics model is three-fold: a) predict the effect of thermal fluctuations on the proposed AFZ mechanism, b) compare thermodynamic model predictions to the BD model, and c) make direct comparison between the Brownian ratchet model (Mogilner and Oster, 1996) and our proposed AFZ model.

Actin Bundling Reaction Kinetics

In modeling the reaction kinetics, as a first approach, we assume pseudo-equilibrium with respect to binding and filament configuration. This implies the movement of the bacterium is slow relative to the reaction kinetics and the movement of the actin filament as a result of bundling. Additionally, we also assume α-actinin concentration is in excess and not significantly depleted in the reaction. Disassembly of an actin bundle would result in re-

supplying α-actinin to the existing free α-actinin pool in the cytoplasm. The binding of each α-actinin molecule to two actin filaments is modeled as a two-step reaction process described by:

$$A + R \underset{k_r}{\overset{k_f}{\rightleftharpoons}} C_1 \tag{4.1}$$

$$C_1 + R \underset{k_2}{\overset{k_1(d)}{\rightleftharpoons}} C_2 \tag{4.2}$$

where A is the concentration of free concentration of α-actinin, R is the density of free binding sites on an actin filament, C_1 is the density of α-actinin molecules bound to only one actin filament, and C_2 is the corresponding number bound to two adjacent actin filaments. The forward and reverse binding rates of the reactions are described by k_f, k_1, and k_r, k_2 respectively. Schematically this is illustrated in Figure 4.3.

To account for the effect of separation distance, d, between two actin filaments in the cross-bridging reaction (Eqn. 4.2), the transition state of the forward rate of the second reaction is modeled as a harmonic spring with activation energy that decreases quadratically with increasing separation distance between the filaments, i.e.

$$k_1(d) = k_{10} \exp \frac{-\gamma_B (d - d_0)^2}{2k_B T} \tag{4.3}$$

where k_{10} is the unstressed binding affinity, γ_B is the bond spring constant, and d_0 is the optimal separation distance for binding.

Based on Eqn. 4.1-2, the evolution of C_1 and C_2 are described by:

$$\frac{dC_1}{dt} = k_f AR - k_r C_1 - k_1(d) R C_1 + k_2 C_2 \tag{4.4}$$

$$\frac{dC_2}{dt} = k_1(d) R C_1 - k_2 C_2 \tag{4.5}$$

The parametric dependence of C_1 and C_2 can be more easily explored if Eqns. 4-5 are made dimensionless. Let $U_1 = C_1/R_T$ and $U_2 = C_2/R_T$ respectively, where $R_T = R + C_1 + C_2$ is the total density of binding sites per unit length along the actin filament. Scaling Eqns. 4-5 to characteristic time as k_r^{-1}, which is the mean time for a singly bound α-actinin to dissociate, the dimensionless equations become:

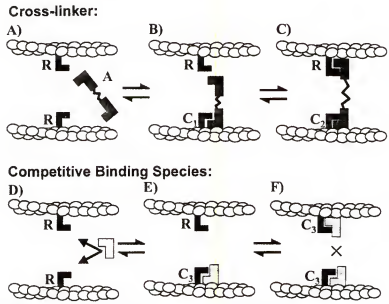

Fig 4.3. In the cross-linker reaction (A-C) α-actinin, A, can bind to free actin-binding sites, R to form both bind singly bound α-actinin, C_1, and doubly-bound α-actinin, C_2. Additionally, a competitive binding species (D-F) can also bind to free actin-binding sites forming, C_3, consequently blocking the formation of cross-bridging structures (C_2).

$$\frac{dU_1}{d\tau} = a(1-U_1-U_2) - U_1 - \alpha\rho(d)(1-U_1-U_2)U_1 + \alpha U_2 \qquad (4.6)$$

$$\frac{dU_2}{d\tau} = \alpha\rho(d)(1-U_1-U_2)U_1 - \alpha U_2 \qquad (4.7)$$

where $\tau \equiv k_r t$ is dimensionless time, $a \equiv Ak_f / k_r$ is the dimensionless α-actinin concentration (or affinity), $\rho(d) \equiv R_T k_1(d)/k_2$ is the dimensionless affinity for the cross-linking second reaction, $\alpha \equiv k_2 / k_r$ is the relative dissociation rates of the two reactions. By

making the Eqns. 4-5 dimensionless, we have effectively reduced the dependence of the model to three parameters: a, α, and ρ.

The equilibrium for U_1 and U_2 can be found from the steady-state solutions to Eqns. 6-7 and are given by:

$$U_{1,eq} = \frac{1+a}{2\rho(d)}\left[\sqrt{1+\frac{4a\rho(d)}{(1+a)^2}}-1\right] \tag{4.8}$$

$$U_{2,eq} = \frac{\rho(d)}{a}U_{1,eq}{}^2 \tag{4.9}$$

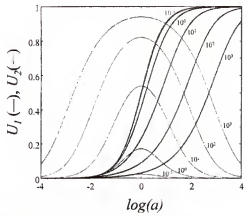

Fig. 4A. Equilibrium reaction kinetics describing as a function of actin filaments versus increasing dimensionless α-actinin concentration, a, as a function of dimensionless crosslinking binding affinity, ρ. U_1 (solid line) and U_2 (dashed line) represent dimensionless concentrations of singly (Fig.2B) and doubly-bound α-actinin molecules (Fig.2C).

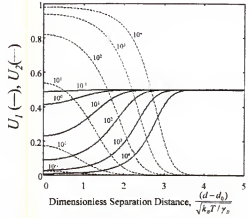

Fig. 4B. Illustration of evolution of U_1 (solid line) and U_2 (dashed line) versus dimensionless separation distance as a function of dimensionless cross-bridging binding affinity, ρ. Increasing the separation distance between two actin filaments results in a significant decrease in U_2 due a reducing ability of α-actinin to span the increasing separation distances.

Shown in Figure 4.4A are plots of U_1 and U_2 versus dimensionless α-actinin concentration, a, for multiple values of the dimensionless cross-linking affinity, ρ. The model predicts that cross-bridging density, U_2, is biphasically dependent on the free α-actinin concentration, with the optimal bundling concentration occurring when α-actinin

concentration is equal to the equilibrium dissociation constant, i.e. at $A=k_r/k_f \equiv K_d$. This relevant result suggests that, although α-actinin is required for bundling and motility, excessive concentrations of α-actinin may in fact inhibit motility. Additionally, because the AFZ mechanism depends on the cross-bridging of actin filaments via α-actinin, it also suggests that the value of the concentration of α-actinin necessary to maximize motility is equal to K_D.

The ability of the cross-bridging reaction to bundle the adjacent filaments depends on the separation distance between the filaments (Eqn. 4.4). The values of dimensionless binding densities, U_1 and U_2 are plotted versus dimensionless separation distance, $(d-d_0)/\sqrt{k_B T/\gamma}$ for various values of $\rho_0 \equiv R_T k_{10}/k_2$ (Fig. 4.4B). From this plot, it is evident that cross-bridging filaments together decreases sharply as the separation distance becomes too large for α-actinin molecules to span the separation distance and form cross-bridging structures to both actin filaments.

Actin Bundling and Competitive Binding

The observation of disrupted actin-cytoskeleton following microinjection of proteolytic α-actinin fragments as previously reported (Dold et al., 1994; Pavalko and Burridge, 1991) is presumably due to competitive binding between endogenous actin-binding domain of α-actinin (Baron et al., 1987; Imamura et al., 1988; Mimura and Asano, 1986) and α-actinin fragments. *In vivo* there may be additionally a host of proteins that can function using similar mechanisms inhibiting bundling of actin, so it is also of interest to investigate how the presence of competitive actin-binding (CAB) species can effect the AFZ mechanism and ultimately force production in the AFZ model.

Here we account for the additional binding reaction of a CAB species represented by following reversible reaction:

$$I + R \underset{k_4}{\overset{k_3}{\rightleftharpoons}} C_3 \tag{4.10}$$

where I represents the CAB, C_3 is the density of CAB molecules singly bound to one actin filament, k_3 and k_4 are the forward and reverse reaction rates respectively. The resulting balance equation is:

$$\frac{dC_3}{dt} = k_3 IR - k_4 C_3 \tag{4.11}$$

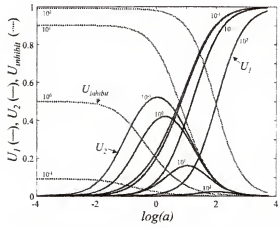

Fig. 4.5. Plot of evolution of U_1 (dotted line), U_2 (solid line) and $U_{inhibit}$ (dashed line) versus increasing concentration of α-actinin, a, for values of $U_{inhibit}$ equal to 10^{-1},10^0,10^1,10^2 and $\rho = 10$, $\alpha = 1$.

Equation 11 can be put in dimensionless form similarly as for C_1 and C_2 by letting $U_{inhibit} = C_3/R_T$ and $g = Ik_3/k_4$, and $\omega = k_4/k_2$. Adding the third reaction component also requires an update in density of free binding sites expression: $R = R_T - U_1 - U_2 - U_{inhibit}$. The dimensionless equations describing the reaction kinetics is then given by:

$$\frac{dU_1}{d\tau} = a - U_1(a + 1 + \rho - \rho U_2 - \rho U_3) + \omega U_2 \tag{4.12}$$

$$\frac{dU_2}{d\tau} = \rho U_1(1 - U_1 - U_2 - U_3) - \omega U_2 \tag{4.13}$$

$$\frac{dU_3}{d\tau} = g(1 - U_1 - U_2 - U_3) - U_3 \tag{4.14}$$

where τ, ρ and a are the same as defined above. The equilibrium for U_1, U_2 and $U_{inhibit}$ can be found from the steady-state solutions to Eqns. 12-14 are given by:

$$U_{1,eq} = -\frac{1}{2\rho(d)}\left[g+1+a-\sqrt{1+g(g+2a+2)+a(a+4\rho(d)+2)}\right] \quad (4.15)$$

$$U_{2,eq} = \frac{1-U_{1,eq}}{1+\dfrac{1+g}{\rho(d)U_{1,eq}}} \quad (4.16)$$

$$U_{inhibit,eq} = \frac{g}{1+g}\left[1-U_{1,eq}-U_{2,eq}\right] \quad (4.17)$$

Figure 4.5 shows U_1, U_2 and $U_{inhibit}$ as a function of increasing α-actinin concentration. This plot shows that the presence of a CAB species can lower the bundling efficiency as expected. Also, the presence of a CAB species increases the required α-actinin concentration for maximal crosslinking.

Formulation of Thermodynamic AFZ Model

For conceptual ease, we will only consider two filaments with two degrees of freedom for movement. Initially the actin filaments will be crosslinked to each other at the distal end and to the existing actin network, which is regarded as providing a rigid support. The free end of the filaments is presumed in to be in contact with a flat barrier representing a cell wall (*in the case of Listeria propulsion*), illustrated in Figure 4.2. The binding of α-actinin to two adjacent nonparallel filaments bundles the filaments by interlocking them locally in a parallel configuration potentially creating a flexural bend in the actin filament which can consequent in pushing force on the load. The energetics to further bundle the actin filaments is provided via the chemical energy associated with the cross-bridging reaction. As the actin filaments are bundled, the two actin filaments, which are not necessarily parallel in orientation, can undergo elastic bending. The bending of an actin filament has an associated mechanical bending energy. The total energy of the system can be described by:

$$E_{total} = \int_0^L \left(E_B(s) + \Delta G_{rxn}(s)\right) ds \tag{4.18}$$

where E_{total} is the total energy of the system, E_B is the energy per unit length along the filament associated with the flexural bending of actin filaments, ΔG_{rxn} is the chemical energy per unit length along the filament associated with the binding of α-actinin and s is the arc length along the filament.

Mechanically the bending energy of the filament is given by the contour integral over the local curvature multiplied by the bending modulus described by (Doi and Edwards, 1986):

$$E_B = \frac{1}{2} B \frac{d^2\mathbf{r}}{ds^2} \tag{4.19}$$

where $B=\lambda_p k_B T$ is the bending modulus r is vector position along the length of the filament The chemical energy of the system can be described by the change in Gibbs free energy:

$$\Delta G_{rxn}(s) = G_{rxn}\left(C_{1,eq}, C_{2,eq}, C_{3,eq}, d(s)\right) - G_{rxn}\left(0,0,0,\infty\right) \tag{4.20}$$

where ΔG_{rxn} represents the change in chemical energy of the system going from infinite separation distance in which there is no binding of α-actinin to a state where the actin filament separation distance is small enough such that α-actinin can bind to actin filaments. The free energy of the system at C_1, C_2, C_3 and a given separation distance between two filaments can be described by:

$$G_{rxn}(C_1, C_2 C_3, d) = R\mu_r + C_1\mu_1 + C_2\mu_2(d) + C_3\mu_3 \tag{4.21}$$

where $R\mu_r$, $A\mu_a$, $C_1\mu_1$, $C_2\mu_2$, $C_3\mu_3$ are the free energies associated with the unoccupied binding sites on actin filaments, unbound α-actinin, number of singly-bound α-actinin molecules/length, number of doubly-bound α-actinin molecules/length bound, and the number of CAB molecules/length bound to actin filaments. The respective chemical potentials for each species is given by:

$$\mu_r = \mu_r^0 + k_B T \ln(R) \tag{4.22}$$

$$\mu_1 = \mu_1^0 + k_B T \ln(C_1) \tag{4.23}$$

$$\mu_2 = \mu_2^0(d_0) + \frac{\gamma}{2}(d - d_0)^2 + k_B T \ln(C_2) \tag{4.24}$$

$$\mu_3 = \mu_3^0 + k_B T \ln(C_3) \tag{4.25}$$

Here, the standard state chemical potentials, μ^0, correspond to a standard state density (1/nm). Note μ_2 is a function of separation distance between two filaments, reflecting a decreasing cross-bridging probability as a function of increasing separation distance as similarly described previously by the forward binding rate constant, k_f. At binding equilibrium, $\Delta G_{rxn}(s)$ becomes:

$$\Delta G_{rxn}(d(s)) = R_T k_B T \ln\left(\frac{R_T - C_{1,eq} - C_{2,eq} - C_{3,eq}}{R_T}\right) \tag{4.26}$$

Increased separation distance beyond the d_0 not only decreases the amount of bundling of actin as noted in Figure 4.4B, but also results in a significant decrease in the magnitude of $\Delta G_{rxn}(s)$ (Fig. 4.6).

Assuming pseudo-equilibrium with respect to binding and filament configuration as described above, the equilibrium configuration of the actin filaments is that which minimizes E_{total}. With both the mechanical and chemical energy of the system described by Eqns. 19-20, the aim is to minimize the integrand of Eqn. 4.18. Mathematically, the task of minimizing the E_{total} is accomplished by applying Calculus of Variations to derive the Euler-Lagrange equation, which is the minimized functional form of the integrand given by:

$$E_B = \int_0^L \frac{B}{2}\mathbf{r}''(s) + \Delta G_{rxn}(\mathbf{r}(s))ds \tag{4.27}$$

Noting the integrand of Eqn. 4.27 is $f(\mathbf{r},\mathbf{r}'')$, the Euler-Lagrange equation describing both the mechanical and chemical energies is then given by:

$$B\frac{d^4\mathbf{r}(s)}{ds^4}+\frac{\partial\Delta G_{rxn}(\mathbf{r}(s))}{\partial\mathbf{r}(s)}=0 \tag{4.28}$$

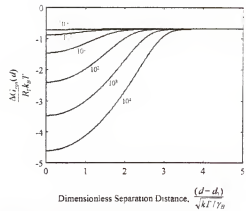

Fig. 4.6. Plot of dimensionless Gibbs free energy versus dimensionless separation distance for various values of dimensionless cross-bridging binding affinity, ρ. Increasing separation distance between actin filaments results in the diminishing ability of α-actinin molecules to form chemical bond linkages and subsequently a decrease in the magnitude the chemical energy of the system.

The four boundary conditions applied are: 1) $\mathbf{r}|_0 = 0$, 2) $\frac{\partial^2\mathbf{r}}{\partial s^2}|_0 = 0$ 3) $\frac{\partial^2\mathbf{r}}{\partial s^2}|_L = 0$, 4) $\int_0^L\left(1+\frac{\partial\mathbf{r}}{\partial s}\right)^{1/2} ds$. Condition 1 reflects two filaments initially bundled together initially at $\mathbf{r} = 0$. Condition 2-3 constraints the slope at $\mathbf{r} = 0$ and $\mathbf{r} = L$ to a linear slope; thus, with no α-actinin zippering, the actin filament is relaxed to an unbent state. Condition 4 is the requirement that the total filament length must equal L.

The equilibrium filament shape is determined by numerically solving Eqn. 4.28. The force exerted per filament is then be calculated by making a small increment along the axis of bundling (horizontal axis). The force generated by the two bundled filaments is given by:

$$\mathbf{F} = -2\frac{dE_B}{d\mathbf{r}} \tag{4.29}$$

Equilibrium Model Force Predictions

Modeling the reaction kinetics as a two-step reaction process and assuming pseudo equilibrium reaction kinetics, the bundling of actin is biphasically dependent on α-actinin concentration (Fig. 4.4A). Mechanically, the bundling of two nonparallel actin filaments

requires the filaments to be interlocked locally along the region that is bundled by α-actinin. This dynamic localized interlocking of filaments due specifically to the bond linkages formed between filaments can cause two nonparallel filaments to align to a collinear configuration similar to a zippering mechanism. Figure 4.2B shows how the AFZ mechanism can generate forces in the formation of cross-bridging bonds between two actin filaments. For filaments in contact with a barrier, such as a cell wall in the case of the actin rocket-like tail associated with Listeria propulsion, the zippering of two actin filaments can result in a flexural bending in the actin filaments consequenting in a pushing force exerted by the actin filaments along the bundling axis (Fig. 4.1). The extent to which actin filaments can be zippered together is dependent on the chemical energy describing the interaction between actin filaments and α-actinin balanced by the mechanical restorative energy resisting flexural bending of the actin filaments described by the mechanical bending energy of the filament (Fig.4.2). Increased bundling of two actin filaments results in not only zippering the two filaments together, but also enhanced flexural bending of the individual actin filaments. Since force is generated through the bundling of actin filaments, one might predict that force would be not only biphasically dependent on α-actinin concentration similarly as U_2 illustrated in Figure 4.4A. Additionally, the optimal bundling efficiency occurring at an α-actinin concentration equal to K_D would be predicted to correspond to the maximal propulsion force.

Minimizing the total energy of the system described by Eqn. 4.18 by numerically solving Eqn. 4.30 as a function of increasing α-actinin concentration, the thermodynamic model predicts propulsion force is biphasically dependent on α-actinin concentration with the maximal propulsion force occurring when the α-actinin concentration is equal to K_D (Fig. 4.7A). This result suggests that propulsion force not only biphasically depends on α-actinin concentration, but also the magnitude of the force generated per filament depends strongly on the mechanical stiffness of the actin filaments.

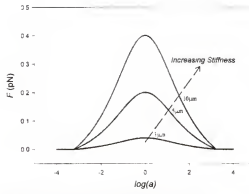

Fig 4.7A. Thermodynamic model force/filament (pN) predictions for the zippering of two actin filaments as a function of α-actinin concentration and mechanical stiffness (λ_P=1, 5, 10 μm). Model parameter values chosen in the simulation: α =20, ε =5, ρ =10, x_c=.99, g =0 R_T =.025nm^{-1}, L=1μm

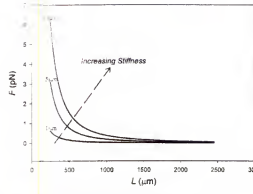

Fig 4.7B. Thermodynamic model force/filament (pN) prediction for the zippering of two actin filaments as a function of filament length and mechanical stiffness (λ_P=1, 5, 10 μm). Model parameter values chosen in the simulation: α = 0.5,α =20, ε =5, ρ =10, x_c=.99, R_T=. 025nm^{-1}

In addition to predicting propulsion force being dependent on the both the reaction kinetics and the mechanical stiffness of the actin filaments, it is also predicted to strongly dependent on the length of the filaments (Fig. 4.7B). The physical reason for this is, assuming the filament is buttressed to the barrier at one end, the bundling of two nonparallel shorter filaments in the direction of the barrier can consequence in a sharper bending moment per unit length compared to two longer filaments. As a result, the AFZ model predicts larger mechanical bending forces associated with the bundling of shorter filaments.

The results presented in Figure 4.7A-B consider only the interaction between actin and α-actinin. Accounting for the presence of a CAB molecule that could potentially bind and block free actin binding sites thus preventing α-actinin from forming cross-bridging bonds could ultimately affect the propulsion capabilities of the AFZ mechanism (Fig. 4.2D-E). To investigate this, we model a CAB species assuming equal binding affinities as the

To account for the short-range interaction between the barrier and the actin filament, we model as a repulsive interaction. Because the underlying physics describing the short-range interaction between the bead and the filament for this model are not well understood, we modeled this interaction using the simplest model described by the fewest parameters: a linear repulsion model with a characteristic repulsion decay parameter, γ_{Wall}. The criterion for choosing the parameter value for γ_{Wall} was to provide both a sufficiently steep repulsion barrier relative to Brownian displacements with the additional constraint that, over the course of one simulation time step, the repulsion force exerted on the bead would not force the bead out of the interacting region so quickly as to cause the simulation to go unstable. For bead displacements near the barrier, a linearly increasing repulsion force is exerted on the interacting bead and is modeled as:

$$F_{Wall,i} = \gamma_{Wall} d \tag{4.40}$$

where d describes the separation distance between the filament and the barrier. To determine if the model predictions depended strongly on this repulsion model, the well-known Leonard-Jones potential describing molecular interactions with a characteristic repulsive term (A/r^{12}) was also simulated. Comparing the time-averaged force between the two repulsion models for a filament 1μm in length and λ_p=10μm, no significant difference in predicted forces was found between the two models.

The parametric dependence of propulsion force, F_{Wall}, as a function of the model parameters can be more easily investigated by making the parameters dimensionless. Taking the characteristic time as D/r_{min}, which is the mean time for a bead to diffuse one length, scaling length by the relaxed spacing between beads $z = d / r_{\min}$, and the deterministic forces, F_i, is scaled by k_BT/r_{min}. The dimensionless form for Equation 4.32 reduces to:

$$dz_i = f_i d\tau_2 + \sqrt{2} dW(\tau_2) \tag{4.41}$$

where f_i is the dimensionless deterministic force and τ_2 is dimensionless time.

The local bending force of the i^{th} bead $F_{B,i}$, is described by:

$$F_{B,i} = -\frac{dE_{B,i}}{d\mathbf{r}_i} \tag{4.33}$$

where E_B is bending energy of the i^{th} bead, and $\mathbf{r}_i$ is the position of the i^{th} bead. Solving Eqn. 4.35 for $F_{B,I}$ results in (Appendix G):

$i=1$:

$$F_{B,1} = -\frac{B}{r_{\min}^3}\left[\mathbf{r}_1 - 2\mathbf{r}_2 + \mathbf{r}_3\right] \tag{4.34}$$

$i=2$:

$$F_{B,2} = -\frac{B}{r_{\min}^3}\left[-2\mathbf{r}_1 + 5\mathbf{r}_2 - 4\mathbf{r}_3 + \mathbf{r}_4\right] \tag{4.35}$$

$2 < i < N$-1:

$$F_{B,i} = -\frac{B}{r_{\min}^3}\left[\mathbf{r}_{i-2} - 4\mathbf{r}_{i-1} + 6\mathbf{r}_i + \mathbf{r}_{i+1}\right] \tag{4.36}$$

$i = N$-1:

$$F_{B,N-1} = -\frac{B}{r_{\min}^3}\left[\mathbf{r}_{N-3} - 4\mathbf{r}_{N-2} + 5\mathbf{r}_{N-1} - 2\mathbf{r}_N\right] \tag{4.37}$$

$i = N$:

$$F_{B,N} = -\frac{B}{r_{\min}^3}\left[\mathbf{r}_{N-2} - 2\mathbf{r}_{N-1} + \mathbf{r}_N\right] \tag{4.38}$$

Chemical bonds formed between beads, modeled as a Hookean springs, can consequently be stretched/compressed away from its optimal separation. The force due to the stretching/compressing of the chemical bond is modeled as:

$$F_{chem,i} = \gamma_B (d_i - d_O) \tag{4.39}$$

where γ_B is spring constant describing the crosslinking bonds, d_i is the distance the chemical bond formed at the i^{th} bead spans cross-bridging two actin filaments and d_O represents the relaxed bond separation distance.

describing this process as similarly described above for a single bound α-actinin molecule, $k_c dt$.

The dynamics of actin filaments are simulated through updating of the three-dimensional displacement of each individual bead along the filament chain. Displacements of each bead are described by:

$$dr_i = \left[\frac{D}{k_B T} F_i\right] dt + \sqrt{2D}\, dW_i(t) \tag{4.30}$$

where dr_i is a unit vector describing the displacement of the i^{th} bead, D is the bead diffusion coefficient, t is time and dW is an increment in three dimensional Weiner process characterized by a Gaussian process with a mean equal to zero and a variance equal to one. Numerically, large Gaussian increments associated with outermost tail regions of the Gaussian curve can cause the simulation to go unstable for a the filament near a steep repulsion barrier. To avoid this potential pitfall, a second-order Gaussian approximation was used (Ottinger, 1996).

The deterministic forces on each bead are described by:

$$F_i = F_{s,i} + F_{B,i} + F_{Chem,i} + F_{Wall,i} \tag{4.31}$$

where F_s is the spring force due to stretching/compression of the springs interconnecting the beads, F_B is the restorative force due to local flexural bending of the actin filament, F_{Chem} is the force due to stretching/compression of the chemical bond interconnecting two actin filaments and F_{Wall} represents the characteristic repulsive force exerted by the wall. The spring force, F_s, is given by:

$$F_{s,i} = \gamma_s (d_i - d_o) \tag{4.32}$$

where γ_s is the spring constant of the springs interconnecting the beads, d_i is the separation distance between the i^{th} bead and an adjacent bead, and d_o is the relaxed spacing between beads (no compression or stretching).

localized segment of the actin filament as a single binding site for an α-actinin. Each bead can exist in one of either three states: 1) free (Fig. 4.3A), 2) bound to a single α-actinin molecule with no cross-bridging bonds formed (Fig. 4.3B) or 3) bound to a single α-actinin molecule forming a crosslinking bond between two actin filaments (Fig. 4.3C). With the exception of the final two beads at the distal end of the filament, initially, the beads are in assumed in State 1. The final two beads along the filament are assumed crosslinked to each other and to the existing actin network providing a rigid support (similar condition applied in thermodynamic model). Beads described by State 1, can bind to α-actinin with an associated binding probability described by $k_f A dt$, where k_f describes the forward binding reaction binding rate of Eqn. 4.1, A represents the free α-actinin concentration, and dt is the time incremented step in the simulation. Beads described by State 2 have three potential possibilities: a) remain in State 2, or transition to b) State 1 or c) State 3. We model this reaction process by first allowing the α-actinin molecule to potentially disassociate at a rate described by $k_r dt$, where k_r is the reverse binding reaction rate described by Eqn 1. If the molecule does not disassociate, the α-actinin molecule bound to the bead can form a chemical bond crosslinking two actin filaments together described by a binding probability $k_{f,2} C_1 dt$ where $k_{f,2}$ is described previously described by Eqn. 4.3. We feel k_{f0}, the binding rate at optimal separation distance, is significantly larger than k_f because the α-actinin molecule can undergo a conformational change and/or geometric positioning upon binding to the actin filament, thus enhancing its binding affinity (Sims et al., 1991; O'Toole et al., 1990), coupled with the fact that k_f additionally accounts for diffusion from the bulk solution, which is not the case in the crosslinking reaction (Fig. 4.3A-C). Thus, we model $k_{f,0}$ as $10k_f$. For bonds interlocking actin filaments together described by State 3, each individual bond between the bead and the α-actinin molecule can also disassociate. We model the rate of disassociation

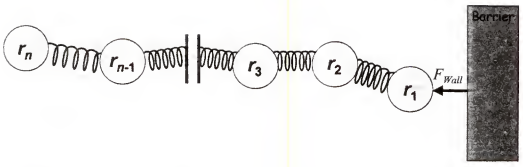

Fig. 4.8. Representation of the Hookean bead-spring chain model used to model Brownian actin dynamics. Beads represent discrete segments along the actin filament. Springs interconnecting the beads can stretch and compress due to both Brownian fluctuations of and deterministic forces exerted on the beads.

springs (Fig. 4.8). The beads themselves do not represent individual monomer segments, rather discrete points along the actin filament. In addition to Brownian fluctuations, each bead is also subject to a set of deterministic forces. Deterministic forces account for chemical interaction, bending of the actin filament, spring force due to compression/stretching of springs interconnecting bead segments and for the short-range interaction between the barrier and the actin filament. The chemical interaction between two adjacent filaments is described in this model by the time-dependent reaction kinetics (i.e., pseudo-equilibrium reaction kinetics is not assumed as in the former model) between neighboring beads. Chemical bonds formed between beads are modeled as Hookean springs that have the capacity to stretch or compress (similar to thermodynamic model).

To characterize the chemical interaction between α-actinin and individual beads, α-actinin can bind to the beads along the filament. This model assumes each bead can bind to one α-actinin molecule. In actuality, each bead represents a segment along the actin filament that could potentially bind a number of α-actinin molecules. However, since the reported average α-actinin density per unit length bound to actin filaments is on the order our modeled bead radius (50nm) (Meyer and Aebi, 1990), we lump the chemical interaction of the

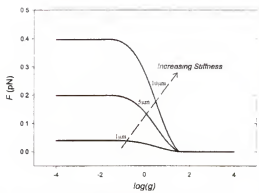

Fig. 4.7C. Thermodynamic model (pN) predictions for the zippering of two actin filaments as function of competitive actin-binding species concentration and mechanical stiffness (λ_p=1, 5, 10 μm). Model parameter values chosen in the simulation: a=.5 α =20, ε =5, ρ =10, x_e=.99, g =0 (no competitive actin-binding species present), R_T =.025 nm^{-1}, L = 1 μm.

α-actinin molecule. The model predicts propulsion force is a decreasing function of increasing concentrations of the CAB species, g. Interestingly the presence of the CAB molecule does not contribute to a significant diminishing effect on the predicted zippering mechanism force for relatively low concentrations. In fact, our model predicts for an α-actinin concentration equal to $.5K_D$ with an assumed CAB species binding affinity equal to α-actinin for actin, the concentration of the CAB species required to significantly decrease the zippering force would be on the order of K_D and higher. The explanation for the deceased propulsion force is due specifically to blocking of free actin binding sites preventing the α-actinin molecule forming bond linkages between two filaments.

Formulation of Brownian Dynamics AFZ Model

A Brownian dynamics (BD) model was formulated based on the classical Hookean bead-spring chain model developed by Rouse (Ottinger, 1996) to investigate how the incorporation of thermal fluctuations of individual actin filaments would affect model force predictions. The Rouse model is based on a linear set of stochastic differential equations of motions to describe the polymer dynamics. The polymer, in this case the actin filament, is represented by a linear chain of N identical spherical beads connected by N-1 Hookean

We model the barrier object as stationary. The force exerted by an actin filament along the bundling axis is then given by the, F_{wall} (ie. the repulsion force exerted by the wall is completely balanced by the pushing force by the actin filament). Due characteristic nature of fluctuating movements of the actin filament near the load, the time-averaged propulsion force is reported and is given by:

$$\langle F \rangle = \frac{1}{t} \int_0^t F_{wall} dt \tag{4.42}$$

where t represents the total simulation time.

Brownian Dynamic Model Force Predictions

The thermodynamic mathematical model describing the AFZ provides a nice tool to investigate not only how the AFZ depends on model parameters such as reaction binding affinities and mechanical properties of actin filaments, but also predicts propulsion forces based on this proposed mechanism. It does, however, omit some important features: thermal fluctuations of actin filaments, multi-filament interactions, and time-dependent reaction kinetics. The thermodynamic model did not account for the Brownian fluctuations of the filaments and the filaments were assumed always in contact with the barrier. In addition, only the bundling of two nonparallel filaments with the assumption of pseudo equilibrium reaction kinetics was considered. In actuality, actin filaments do undergo Brownian fluctuations (Kas et al., 1996), there are a number of actin filaments involved in the propulsion mechanism as in the case of *Listeria monocytogenes* (Sechi et al., 1997), and the reaction kinetics may not attain equilibrium as assumed in the former model. In this paper we will focus specifically on the effect of thermal fluctuations of individual actin filaments and time-dependent reaction kinetics on the predicted propulsion forces. Multi-filament and filament orientation/distribution will be considered as separate issues in future work.

Incorporating these aspects, a BD model based on the classical Rouse bead-spring chain model (Ottinger, 1996) was formulated (Fig. 4.8). To allow the filament to bend and compress due to thermal fluctuations, the value for the spring constant of the springs interconnecting the beads along the filament was chosen such that displacements due to thermal energy represented a value ≈5% of the relaxed spacing between the beads, r_{min}. The time increment in the simulation was chosen such that Brownian displacements of individual beads, on the order of $dt^{1/2}$, were approximately $.01r_{min}$.

To investigate how incorporation of Brownian fluctuations affects propulsive force, actin filaments near a barrier were simulated. Accounting for Brownian fluctuations of the individual actin filaments near a barrier and assuming no chemical interaction between individual filaments (i.e. no zippering mechanism), the BD model predicts the time-averaged propulsive force, $\langle F \rangle$, as a linear function of increasing mechanical stiffness, λ_p (Fig. 4.9A). These results suggest that flexural properties of actin filaments coupled with the thermal fluctuations of individual filaments can result in the generation of significant time-averaged propulsion forces. This set of conditions is analogous to Mogilner and Oster's previous Brownian ratchet model (Mogilner and Oster, 1996). Based on these results, our data supports their hypothesis that thermal fluctuations of actin filaments can significant forces.

Accounting for the chemical interaction between two filaments the BD model predicts $\langle F \rangle$ depends strongly on the free α-actinin concentration, a, in addition to the mechanical stiffness of the actin filaments (Fig. 4.9B). Further, the predicted increase in $\langle F \rangle$ is specifically due to the bundling of two actin filaments via α-actinin as similarly predicted by the thermodynamic model (Fig. 4.7A). These results also suggest that accounting for the Brownian fluctuations of actin filaments near a barrier results in a lower *time-averaged* propulsion force compared to the equilibrium propulsive force predicted by the thermodynamic model. The difference between the two model predictions is primarily

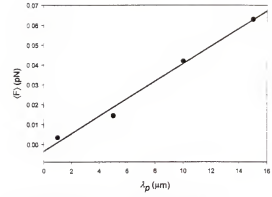

Fig. 4.9A.. Brownian dynamic model prediction of time-averaged force/filament, $\langle F \rangle$ (pN), as a function of increasing mechanical stiffness. No chemical interaction between adjacent filaments was considered here. Model parameter values chosen in simulation: $dt = 2.3e^{-6}$s, $\gamma_s = .12$pN/nm, $\gamma_{Wall} = 1$pN/nm, $L = 1\mu$m, iterations = $1x10^6$.

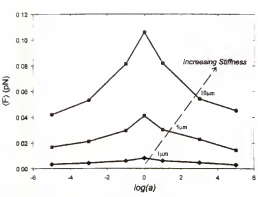

Fig. 4.9B. Brownian dynamic model prediction of time-averaged force/filament, $\langle F \rangle$, as a function of increasing dimensionless α-actinin concentration for various persistent length values (λ_p=1, 5 10μm). Model parameter values chosen in simulation: $dt = 2.3e^{-6}$s, $\gamma_s = .12$pN/nm, $\gamma_{Wall} = 1$pN/nm, $L = 1$ μm, iterations = $1.5x10^6$.

because, in the BD model, the filament can undulate away from the barrier to a position in which the repulsive interaction force exerted on the actin filament by the barrier is negligible. The result is a lower predicted time-averaged propulsion force compared to the thermodynamic model force predictions.

Discussion

Although experimental evidence supports the notion that many cellular movements do not require molecular motors, but instead are largely determined by the actin-containing cytoskeleton, grasping a fundamental understanding of mechanism(s) involved has been challenging. Long-standing arguments considering the free energy released in the polymerization of actin as a possible mechanism of propulsion provide little, if any, foresight describing the transduction of chemical energy into mechanical force. More recently, using

statistical mechanical arguments Mogilner and Oster (Mogilner and Oster, 1996) proposed that the thermal fluctuations of the free end of actin filaments supported by an existing crosslinked actin network could create a pushing force. In their model, the crosslinked actin network is regarded as a rigid anchor, however, their model does not account for mechanochemical forces involved in the formation of highly oriented actin filament networks via crosslinking proteins as experimentally observed in the actin rocket-like tails of Listeria. In this paper we explore an independent and potentially complementary mechanism: the interaction between actin filaments and actin crosslinking proteins. Taking into account the *in vivo* function of actin-binding proteins (Ayscough, 1998), here, we have analyzed the interaction between actin filaments and actin-bundling proteins and demonstrated that this dynamic and regulated interaction can give rise to significant cellular forces.

In the analysis presented, we find the AFZ mechanism, using both the thermodynamic and BD modeling arguments, the bundling of two nonparallel actin filaments can generate significant pushing forces. One important aspect is that the bundling of actin filament is biphasically dependent on the α-actinin concentration with the maximal bundling efficiency occurring at an α-actinin concentration equal to K_D. Based on this property of bivalent molecules, one key prediction from both models is that mechanically the pushing force exerted by an actin filament is predicted be biphasically depended on α-actinin concentration, and linearly dependent on the mechanical stiffness. Our model also predicts larger mechanochemical pushing forces associated with shorter filaments as compared to longer one. Based on our BD model predictions, our results support the concept that Brownian fluctuations of actin filaments as previously described the by Brownian ratchet model by Mogilner and Oster (Mogilner and Oster, 1996) can generate pushing forces. Lastly, we find that, based on BD model predictions, that additionally accounting for the

specific interaction between actin filaments and α-actinin can result in a significant increase in the pushing forces exerted by actin filaments.

As described by the AFZ model, mechanical force is generated through the dynamic formation of chemical bonds interlocking two actin filaments together described by a zippering mechanism. In order for this mechanism to be potentially plausible in describing the intracellular motility of Listeria, one key implicit assumption is that the α-actinin bundling velocity, V_B, is on the order of Listeria speeds. Although the velocity of Listeria can be variable, a reasonable value is on the order .15μm/s. Accurately measuring the intracellular free α-actinin concentration can be difficult, but a reasonable value we feel is on the order of $3x10^{-7}$M. Modeling the reaction kinetics describing the bundling of actin via α-actinin as a two-step reversible reaction (Eqns. 4-5) and assuming an α-actinin concentration ≈3e-7M, and k_{fs1} ≈6.8e5 $(Ms)^{-1}$ we find for an maximal $V_B \approx 7$nm/s (Appendix H) which is significantly greater than the experimentally measured Listeria speed, suggesting the bundling reaction can potential occur at speeds on the order of speeds experimentally measured. These results suggest that motility of Listeria is reaction-rate limited. That said, this should not be interpreted to suggest that the Listeria bacterium is moving necessary at "top speed", but rather, the bacterium is moving at or near its maximal speed at the assumed free α-actinin concentration ($3x10^{-7}$M). Increasing the α-actinin concentration beyond this value, but less than or equal to K_D would be predicted to result in an increased speed. Future work includes, investigating a dose-response for Listeria speed versus α-actinin concentration. These results also potentially provide key insights in elucidating previous experimental observations by Dold et al. (1994) that microinjection of proteolytic α-actinin fragments in Listeria-infected cell results in a complete inhibition Listeria motility. Similar to the analogy of a chain and sprocket, removing the teeth along the sprocket prevents force transduction from the sprocket to the chain; similarly, blocking of free actin-binding sites by

the α-actinin fragments would prevent the bundling of actin and the transduction of mechanochemical forces involved in the aligning of actin filaments

In conclusion, we expect the AFZ model describing the transduction of chemical energy into mechanical force can be used to potentially describe a number additional cellular movements such as intracellular motility of vaccinia virus (Cudmore et al., 1995) and other bacterial pathogens such as *Shigella flexneri* and *Rickettsia conorii* (Gouin et al., 1999) along with membrane filopodia lamellipodia extension. Related to actin-based motility as in former examples, the nematode sperm relies exclusively on the major sperm protein for locomotion (Roberts and Stewart, 2000). Here, filament bundling of the major sperm protein in the formation of cellular protrusions does not appear to involve additional bundling proteins, but instead may potentially involve a similar filament bundling mechanism through lateral interactions between the individual fibers. Finally, in the case of DNA replication, the helical winding of DNA strands following the DNA polymerase at the replication fork is reminiscent of a similar zippering mechanism in which attractive hydrophobic interactions between the individual DNA chains energetically drives the zipping mechanism.

CHAPTER 5
A QUANTITATIVE STUDY RELATING THE INTRACELLULAR PROPULSION OF *Listeria monocytogenes* AND ALPHA-ACTININ CONCENTRATION

Introduction

A number of cellular movements such as membrane filopodia and lamellipodia extension and the intracellular motility of *Listeria monocytogenes* (Tilney and Portnoy, 1989) and *Shigella flexneri* (Zeile et al., 1996) occur independent of molecular motors, instead directly involve the coordinated assembly of the cellular cytoskeleton. One such example, the actin-based motility of *Listeria monocytogenes*, presents a dynamic and relatively simplistic motility model compared to the more highly integrated motility system of tissue cells. The actin-based motility of Listeria affords the advantage to more directly study both the dynamic and regulated interactions of a number of cytoskeleton components and how these components can generate significant cellular forces. *Listeria monocytogenes* are pathogenic rod-shaped gram positive ubiquitous bacteria that can be phagocytosed by tissue cells. Once phagocytosed, the bacteria grow and proliferate in the cytoplasm of the host cell. Facilitating the cell-to-cell spreading, the bacterium recruits, polymerizes and assembles host cytoskeleton proteins at one pole of the bacterium forming an actin rocket-like tail composed of actin filaments. The actin tail functions to continuously propel the bacterium to the host cell membrane forming long filopodia-like protrusions that ultimately can be ingested by neighboring cells and the infection process begin again in the new cell (Southwick and Purich, 1996). It is well established that the actin tail at the rear of the bacterium is composed largely of actin filaments. In addition to actin filaments, α-actinin, an actin bundling protein,

has been found to be in high concentrations distributed along the actin tail. (Dabiri et al., 1990; Nanavati et al., 1994; Temm-Grove et al., 1994; Sechi et al., 1997). Alpha-actinin is a dumbbell shaped protein with molecular weight ≈100kDa possessing one actin filament-binding domain at each end of the molecule (Suzuki et al., 1973; Vandekerckhove, 1990). Accordingly, each α-actinin molecule can form cross-bridging chemical bonds between two actin filaments interlocking the filaments in a parallel configuration. Experimentally, it has been previously been demonstrated that a randomly oriented actin network, when exposed α-actinin, can undergo a dynamic rearrangement of the actin filaments form collinearly aligned actin filament bundles (Wachsstock et al., 1993; Bartles, 2000).

Several lines of evidence suggest the interaction between actin filaments and α-actinin plays a critical role in the intracellular propulsion of *Listeria monocytogenes*. First, previous microinjections in Listeria infected cells by Dold et al. (1994) of the 53kDa proteolytic α-actinin fragments resulted in complete arrest of Listeria. This was presumably due to competitive binding between actin-binding domain of endogenous free α-actinin and the α-actinin fragments. Second, high resolution electron microscopy imaging by Sechi et al. (1997) show the physical re-arrangement of the actin filament network from randomly oriented near the bacterial cell wall to increasingly collinearly aligned actin bundles with increasing distance from the rear of the bacteria. Further, experiments by Sechi et al. (1997), show this dynamic alignment of the actin filaments is due to crosslinking by α-actinin. Third, using the reconstituted *in vitro* actin-based assay to reconstitute sustained movement of Listeria, recently Loisel et al. (1999) reported that incorporating α-actinin in the portfolio of purified proteins in the reconstituted assay resulted in more efficient cellular movements of the propulsion of Listeria.

In this present work, a quantitative study relating α-actinin concentration directly to the intracellular speed of *Listeria monocytogenes* is presented. Intracellular concentration of

α-actinin is adjusted by microinjecting purified concentrations of α-actinin into Listeria infected cells. The motile speed of the bacteria is calculated by recording the time-incremented positions prior to and following the microinjection of α-actinin. A key result of this work is a dose-response curve relating bacteria motile response to increasing α-actinin concentration. To our knowledge this is the first quantitative study relating Listeria speed to intracellular α-actinin concentrations. These results underscore the importance of the actin bundling protein α-actinin and its role in actin-based cellular movements.

Materials and Methods

Purification of Alpha-Actinin

One hundred grams of chicken gizzard were cleaned of fat and connective tissue. After weighing cleaned gizzards they were suspended in 10 volumes of cold distilled water containing 0.5 mM PMSF and homogenized in a Waring Blender using 3X10 sec. bursts on high setting. The suspension was centrifuged at 7000 rpm for 10 minutes at 4°C in a Beckman JA10 rotor. The supernatant was discarded and the pellet resuspended in 10 volumes of cold distilled water containing 0.5 mM PMSF and blended for 10 sec on low setting. The suspension was centrifuged at 7000 rpm for 10 minutes at 4°C in a Beckman JA10 rotor. The supernatant was discarded and the pellet was resuspended in 10 volumes of Buffer A (2 mM Tris, 1mM EGTA, 0.5mM PMSF, pH 9.0 at R.T.). The suspension was stirred gently at 37°C for 30 minutes. The suspension was centrifuged at 7000 rpm for 10 minutes at 4°C in a Beckman JA10 rotor. The pellet was discarded and the supernatant was brought to pH 7.0 with 0.5M acetic acid. 1 Molar $MgCl_2$ was added to make the solution 10mM with respect to $MgCl_2$. The solution, now cloudy was allowed to stir for 15 minutes at room temperature. The suspension was centrifuged at 7000 rpm for 10 minutes at 4°C in a Beckman JA10 rotor. The pellet was discarded and the supernatant was placed on ice and

14.9 g of $(NH_4)_2SO_4$ per 100 ml of supernatant were added with stirring. The minor precipitate was removed by centrifugation at 7000 rpm for 10 minutes at 4°C in a Beckman JA10 rotor. A further 5.6 g of $(NH_4)_2SO_4$ per 100 ml of supernatant were slowly added with constant stirring for 1 hour. The suspension was centrifuged at 7000 rpm for 10 minutes at 4°C in a Beckman JA10 rotor. The supernatant was discarded and the pellet resuspended in 5 ml of Buffer B (20 mM NaCl, 0.1 mM EDTA, 15 mM 2-mercaptoethanol, 20 mM Tris/acetate, pH 7.6). The suspension was dialyzed against Buffer B three times at 100 volumes. After dialysis the suspension was centrifuged at 40,000 rpm for 10 minutes at 4°C in a TLA 100.3 rotor. The pellet was discarded and the supernatant loaded onto a 5 ml HiTrap Mono-Q column equilibrated with Buffer B. The proteins were eluted with a 20 volume linear gradient of Buffer B to Buffer B with 1.0 M NaCl. The peak of interest was precipitated with 30 grams of $(NH_4)_2SO_4$ per 100 ml at 4°C with constant stirring for 1 hour. The suspension was centrifuged at 9000 rpm for 10 minutes at 4°C in a Beckman JA20 rotor. The supernatant was discarded and the pellet resuspended in minimum amount of Buffer B and chromatographed on a Superdex 200 column (26mm X 70cm). A single peak of α-actinin is eluted which is > 95% pure by SDS electrophoresis.

Tissue Culture and Infection Protocol

The PtK2 cell line (derived from the kidney epithelium of the kangaroo rat *Pororous tridactylis*) was seeded onto a 35mm tissue culture dish at a concentration of 1e6 with 3ml of culture media (MEM with added 10% (volume basis) fetal calf serum, 1% (volume basis) penicillin-streptomycin) and incubated in a cell culture incubator at 37°C and 5% CO_2. *Listeria monocytogenes* wild-type strain 1043S was grown overnight at 37°C in brain heart infusion broth (Difco, Detriot, MI). The bacteria were harvested at mid-log growth phase, washed and centrifuged three times in phosphate buffer solution (PBS), and re-suspended to a

concentration of ≈ 10^7cells/ml in cell culture media (MEM, 10% fetal calf serum). 2.5e5 bacteria cells were then added to each culture dish containing 3ml culture media (*antibiotic-free media*) followed by 10 minutes of centrifugation of each culture dish at 400x *g* at room temperature to facilitate the infection process. Following centrifugation, cells were incubated at 37°C and 5% CO_2 for 45 minutes. After incubation period extracellular bacteria were removed from the culture dishes by washing three times with PBS and replacing with 3ml of culture media containing gentamicin sulfate (10μg/ml) to prevent extracellular growth of bacteria. Culture dishes were then placed in incubator at 37°C and 5% CO_2 for 1-4 hours during which the microinjection experiments took place.

Video Microscopy and Microinjection

Experiments were performed using a Nikon Diaphot inverted microscope equipped with a charged couple device camera and viewed using 63x objective lens. The stage microscope temperature was controlled at 37°C using a MS-200D perfusion microincubation system (Narishige, Tokyo). Individual PTK2 cells were microinjected using the Eppendorf 5242 microinjection system and to precisely position microneedle for the injection, the Eppendorf 5171 micromanipulator was used as previously described by Southwick and Purich (Southwick and Purich, 1994). The time-lapse movement of motile Listeria was recorded for a period of 4 minutes with a 20s time increment between recorded frames by capturing and storing digital images from CCD camera directly to computer hard drive. Microinjection of the purified α-actinin was performed after 3-4 time-lapse displacements of motile bacteria.

Data Analysis

Following microinjection experiments, the velocities of motile Listeria prior to and following the microinjections was analyzed using the image processing software Optimas (Media Cybernetics, Silver Springs, MD). A computer program using Optimas' programming language, Analytical Languages for Images (ALI), was written that both located the two-dimensional center of mass of individual bacterium based on image contrasting, and stored the updated time-incremented position of the bacterium to an output file in an automated fashion. Two to four motile bacteria within each microinjected cell was analyzed. To allow the microinjected cell to equilibrate following the microinjection, the 40s immediately following the microinjection was not used in the data analysis. To characterize the dose-response of the bacteria to α-actinin concentration, the ratio of the mean speed prior to and post injection was calculated given by:

$$\Re = \frac{\langle S_{Pre} \rangle}{\langle S_{Post} \rangle} \tag{5.1}$$

where $\langle S_{Pre} \rangle$ and $\langle S_{Post} \rangle$ are the mean speed prior to and post injection respectively. To determine if the microinjection of α-actinin had a significant effect on the motile response of bacteria, a one-tailed statistical test was used: H_o: $\Re = 1$ (*no significant difference due to injection*) with a 5% level of significance (type I error). P-values from respective significant tests are reported.

Results

Approximately two hours following the Listeria infection of PTK2 cells, the bacteria would begin actively forming an actin rocket-like tail continuously propelling the cell forward. Experimentally the actin tail can be easily observed as a growing phase dense tail immediately behind the propelling bacterium. Intracellular motile speeds of Listeria can be

quite variable, ranging typically ranging from .05-.15μm/s (Sanger et al., 1992; Dabiri et al., 1990; Southwick and Purich, 1994). The variability in the speeds is primarily regarded to be due to differences in age of tissue culture cells at the time of the infection. Other factors such as phenotypic differences between bacteria and the ratio of (bacteria concentration)/(cell culture monolayer coverage) may also be contributing factors. To minimize these potential differences among microinjection treatments, the motile response was calculated by comparing the pre- and post-treatment motile response among the same cells.

Once the Listeria began to actively assemble the actin rocket tails, purified α-actinin concentrations was injected in cytoplasm of the PTK2 cells associated with actively migrating Listeria. The motile response to a microinjection treatment was obtained by recording the time-lapse movement of the motile Listeria using an automated video microscopy system. The time increment between recorded images was 20s with and the total recorded time per microinjection treatment of three minutes. Multiple microinjection treatments were performed at each α-actinin concentration. The motile response of an individual Listeria was calculated by extracting the time-incremented displacements from the digitally recorded images in an automated fashion using image analysis software and a custom-written cell tracking algorithm. The motile response of each individual Listeria was characterized as the ratio of mean cell speed prior to and following the microinjection as described by Eqn. 5.1. Cells receiving the same microinjected treatment concentration of α-actinin were pooled together to calculate an average population treatment response.

A series of four microinjected α-actinin concentration experiments was investigated with a microneedle concentration ranging 1.5μM- 17μM. Assuming a) an injection volume of 10% of total host cell volume (Dold et al., 1994), b) the nominal value for the intracellular α-actinin concentration the host cells to be 0.3μM (personal communication with Daniel L. Purich), and c) assuming the free α-actinin concentration is not significantly depleted

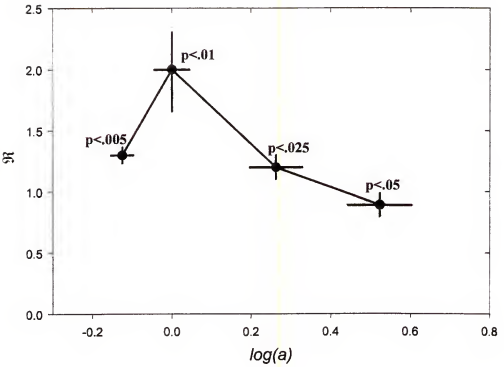

Fig. 5.1. Plot in dose-response of Listeria motility to increasing dimensionless concentrations of actinin, *a*. Dimensionless actinin concentration is obtained by scaling estimated intracellular actinin concentration to the disassociation constant ($K_D \approx 0.6\mu M$ (Wachsstock et al., 1993)). For the calculation of total intracellular concentration a total injected volume of 10% was assumed along with nominal cytoplasm concentration of actinin prior to injection of 0.3μM. Horizontal error bars represent at 5% range for total volume of actinin injected. Vertical error bars represent ± standard error. P-values represent significant level for the null hypothesis: $H_0>1$ (for injected actinin concentrations: 1.5, 3, 8μM) or $H_0<1$ (for actinin concentration =17μM). Significant p-values suggest measured response positively/negatively affected the motility compared to prior to the microinjection.

during the formation of the actin tail, an estimate of the total α-actinin was obtained. Scaling the estimated total α-actinin concentration (post injection) by the disassociation constant for α-actinin, $K_D \approx 0.6\mu M$ (Goldmann and Isenberg, 1993; Wachsstock et al., 1993), the dose response to increasing concentrations is illustrated in Fig. 5.1. These data strongly suggests the motility of Listeria is biphasically dependent on the α-actinin concentration with a

maximal cell speed at an α-actinin concentration equal to the disassociation constant. An α-actinin concentration equal to the disassociation constant represents the concentration when half of the total actin filament binding sites for α-actinin are occupied bound to an α-actinin molecule. Increasing the α-actinin concentration to a value greater than half the binding sites being occupied results in the motility is characterized as a decreasing function of increasing α-actinin concentration. These results are suggestive that the role of actin filament bundling via α-actinin is critical to the intracellular propulsion mechanism of Listeria.

Discussion

Through both tradition biochemical techniques and genetic approaches the actin-based motility of *Listeria monocytogenes* has been dissected revealing the major biochemical proteins involved in the formation of the actin rocket tail. Based on these studies, the actin-based cellular movements of Listeria involves four basic functioning proteins: 1) an actin nucleation protein (e.g. Arp 2/3 complex) functioning to stimulating actin assembly, 2) actin depolymerizing protein (e.g. cofilin) and capping protein maintaining a steady supply of monomeric actin, 3) monomeric actin sequestering proteins (e.g. profilin) aiding in the rapid localized assembly of actin and 4) actin bundling protein (α-actinin) serving to form oriented actin filament bundles (Dold et al., 1994; Loisel et al., 1999; Sechi et al., 1997; Southwick and Purich, 1996).

Among these proteins, less is specifically known about the exact role α-actinin plays in this propulsion mechanism. Recently, using the reconstituted actin-based motility assay, Loisel et al. (1999) reported α-actinin was not required for motility; however, its presence did result in more efficient and effective cellular movements. In contrast to the reports of a

diminished role of α-actinin in Listeria propulsion, in the case intracellular motility of Listeria, Dold et al. (1994) demonstrated α-actinin is required for bacterial cell motility.

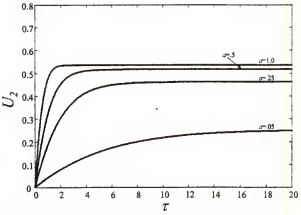

Fig. 5.2. Plot of dimensionless concentration of crosslinked actinin, U_2, as a function of dimensionless time, τ, for various dimensionless actinin concentration, a. Reaction kinetics based on Eqn 4.1-2 and 4.6-7. Parameter values assumed: α=100 and ρ=10, dimensionless bond stiffness and crosslinking binding affinity respectively.

Further supporting the significance of α-actinin in propulsion of Listeria is the increasing amount of evidence high concentrations of α-actinin localized along the lengths of the actin tail (Dabiri et al., 1990; Nanavati et al., 1994; Temm-Grove et al., 1994; Sechi et al., 1997) . Based in part on this evidence, we feel the reconstituted motility assays are not wholly representative of the authentic *in vivo* environment experienced by the Listeria. Another concern with regard to the *in vitro* assay is the fact that the effective cytoplasmic viscosity is inside the host cell may be as high as two orders of magnitude greater than in the *in vitro* reconstituted assays resulting in greatly larger drag forces exerted on the motile Listeria (Bausch et al., 1999). Interestingly related to this, the bacteria cells speeds measured in reconstituted assays are 3-4 times less than typical values reported for the intracellular motility of Listeria (Loisel et al., 1999). These differences pointed out here suggest significant differences may exist between the *in vitro* motility assay and *in vivo*. The aim of this study was to quantitatively access the role of α-actinin in the intracellular motility of Listeria. Using purified concentrations of α-actinin, the dose-dependent relationship of the motile response of Listeria to increasing intracellular α-actinin concentrations was

determined. These results suggest the intracellular speed of Listeria is a biphasic function of α-actinin concentration. An increased concentration of α-actinin from basal levels up to, but not beyond, a concentration equal to the dissociation constant is predicted effect the bundling reaction kinetics between actin filaments and α-actinin via two important ways. First, we model the α-actinin bundling reaction kinetics as a two-step reversible reaction as described by Eqn 4.1-2. One important effect of increasing the concentration of α-actinin is to increase the increased bundling velocity or the rate of forming crosslinks between actin filaments (Fig. 5.2). Increasing the rate at which actin filaments are bundled would be expected to cause an increased rate of filament re-alignment in the actin network near the rear of the Listeria. Recall from Chapter 4, we demonstrated the bundling of nonparallel filaments buttressed at one end near a barrier (such as a cell wall in the case of Listeria) can consequent in a pushing force on the barrier. We called this process the "Actin Filament Zipper" (AFZ). In the case of an interconnected network of actin filaments such as exhibited in the case of the actin tails of Listeria (Sechi et al., 1997), increasing the bundling velocity of actin filaments from a random orientation to a collinearly aligned network would be predicted by the AFZ model to result in a faster forwardly moving zipper. This is consistent with our experimental data. In addition to increasing the bundling velocity, increased concentrations, up to a concentration equal to the disassociation constant, would result in a more efficient bundling of actin filaments by increasing the density of cross-bridging bonds. Increasing the density of crosslinks is predicted, according to the AFZ model, to generate significantly greater propulsive forces. Increased propulsive forces would be required for higher cell speeds to offset increases in the effective drag force exerted on the motile bacteria.

Although our data does strongly suggest that intracellular motility is a function of α-actinin concentration, a number of assumptions used in this analysis should be further validated. One implicit assumption is that in the microinjected α-actinin protein can actively

bind to and form cross-bridging bonds between actin similarly as endogenous αactinin. It is worth noting the source for the purified α-actinin is from a muscle cell, while the host is a non-muscle cell. That said, it has previously been documented, that the same gene probably encodes both the muscle and non-muscle isoforms of α-actinin, with the main difference between the two isoforms being attributed to the non-muscle isoform being Ca^{2+} sensitive (Vandekerckhove, 1990). The validity of this assumption could be tested by fluorescently labeling α-actinin prior to the being injected and quantifying the fluorescence emission distribution to determine if it binds to actin filament comprising the actin tail. A second assumption invoked is volume of α-actinin solution injected is taken as ≈10% of total cell's volume. This is a general assumption often used in these types of studies, however, because the microinjection system uses a constant flow mechanism for delivery of injected proteins, the actual volume of injected protein is a function of both length of time microneedle is inside cytoplasm of cell and the microneedle protein profusion rate. Variability in the injection volume could be analyzed using fluorescently labeled α-actinin or another protein such as bovine serum albumin (BSA), microinjecting a population of cells and measuring the fluorescence emission associated with each injection (volume is proportional to fluorescence emission). In the data analysis, the horizontal error bars correspond to a 5% range of injected volume around the assumed mean value of 10% (Fig. 5.1). Lastly, we assumed a cytoplasmic α-actinin concentration of 0.3μM in calculating the total cytoplasmic α-actinin concentration. A better approach would be to actually measure the mean α-actinin concentration of cultured PTK2 cells using a technique such as western blot analysis.

In summary, this is the first quantitative study investigating the relationship of the intracellular motility of Listeria to α-actinin concentration. We found increasing the intracellular concentration to a value equal to the disassociation constant yielded increasing mean cell speeds with the maximal cell speed at an α-actinin concentration equal on the order

of the disassociation constant. Conversely, increasing the concentration of α-actinin for values greater than the disassociation constant resulted in motility as a decreasing function α-actinin concentration. Together, these results suggest α-actinin is a critical component in the intracellular propulsion of Listeria. Additionally, considering the *in vivo* function of α-actinin and the interaction between actin filaments and α-actinin, we believe, the biphasic dose-dependency of speed on α-actinin concentration can be potentially be explained via an actin filament zipper mechanism as proposed in Chapter 4. Here the bundling of actin filaments via α-actinin crosslinks can generate propulsive forces for Listeria motility through the dynamic zippering of actin filaments to a collinearly aligned actin filament network. Enhancing or reducing the density of these crosslinks would be predicted, according to the AFZ model, to ultimately affect the Listeria speed. Finally, though more experiments should be investigated, preliminarily these results are consistent with an actin filament zipper mechanism as a propulsion mechanism utilized by *Listeria monocytogenes*.

CHAPTER 6
SUMMARY AND FUTURE DIRECTIONS

Summary

Active cell motility, be it eukaryotic or prokaryotic cells, is critically important in a number of physiological and pathological processes such as wound repair and the spreading of infectious diseases. Here we have investigated two important areas of cell motility: 1) adhesion-mediated tissue cell migration, 2) the actin-based mechanochemical propulsion of *Listeria monocytogenes*.

In our first study, related to adhesion-mediated cell migration, using a novel three-dimensional collagen-based cell migration assay developed by Myles et al. (2000), we investigated the relationship between the adhesiveness of the cell-matrix and cell motility. Controlling the adhesive properties of the assay, cell migration was found to be biphasically dependent on the adhesiveness of the substratum with maximal motility sustained around intermediate levels of adhesiveness. Using a mathematical model describing adhesion-mediated cell migration, maximal displacements, interestingly, did not correlate to maximal cell speed as previously reported for two-dimensional cell migration (DiMilla et al., 1993; Duband et al., 1991; Huttenlocher et al., 1996; Keely et al., 1995; Palecek et al., 1997), but rather to increased cell persistence time and cell persistence length (Burgess et al., 2000). Increasing the level of adhesiveness greater than intermediate levels, resulted in both reduced migration and a more tortuous cell path. These results suggest for three-dimensional migration providing optimal level of adhesiveness correlates to an increased ability of the cell to maintain a constant directional polarity resulting in increased migration rates. Together,

we feel our results emphasize potentially different migration strategies utilized by cells in the more highly complex three-dimensional environment compared to migration on planar surfaces and may involve alternate and/or more complex biochemical processes such as the up-regulation of metalloproteinases (Deryugina et al., 1998; Haas et al., 1998).

Building upon our previous study suggesting that by varying the adhesiveness of the matrix, one could potentially modulate different cell migration responses, we developed a three-dimensional RGD adhesion gradient assay to investigate if directed cell migration could result by controlling the adhesive properties of the matrix. Varying the adhesive gradient steepness, we tracked and analyzed the time-incremented displacements of multiple cells within the RGD-gradient collagen assay and found a small but statistically significant biased migration of cells towards increasing gradient concentration. Contributing to this biased migration was a biased orientation of cells, termed *topotaxis*, towards increasing gradient concentration. Although admittedly meager biased migration rates were measured, this study is particularly significant, in that, it is the first to both directly observe and statistically measure three-dimensional haptotaxis migration.

In the second study, motivated by experimental evidence demonstrating the both the association and the dynamic reorganization of actin filaments was due to the actin bundling protein α-actinin in the formation of the actin-rocket like tail of *Listeria monocytogenes*, we investigated the dynamic and regulated interaction of actin filaments and the actin bundling protein α-actinin in the context of how this interaction could potentially play a role in the intracellular propulsion mechanism utilized by Listeria. We investigated this interaction by formulating both a thermodynamic and Brownian dynamic (BD) model. Our models accounted for the bundling reaction kinetics of actin filaments via α-actinin, cross-bridging bond stiffness, and the mechanical properties of actin filaments. A key prediction from the thermodynamic model was that the force/filament generated due to the zippering of actin

filaments was a biphasic function of α-actinin concentration, with maximum force/filament predicted to occur when the free α-actinin concentration was equal to K_D. Accounting for the Brownian fluctuations of individual actin filaments, two key results based on the BD model are noted here. One, our BD model predicts that simply accounting for the flexural properties of actin filaments coupled with the thermal fluctuations of individual filaments can result in the generation of significant time-averaged propulsion forces for the thermal fluctuations of actin filaments near a wall can generate significant cellular forces. These results support the previous hypothesis of a Brownian ratchet put forth by Mogilner and Oster (1996) in that thermal fluctuations of actin filaments can generate forces. Secondly, we learned that by additionally accounting for the interaction between actin filaments and α-actinin our model predicts significantly greater cellular forces could be generated due to the bundling of actin filaments. Together, our models predict the interaction between actin filaments and α-actinin can lead to a dynamic re-aligning of actin filaments to a collinear configuration as similar observed in the actin rocket tails of Listeria, and this re-aligning process can generate significant cellular forces. We called this process the "Actin Filament Zipper".

Finally, interested in experimentally testing our theoretical predictions for the dependence of force/filament as a function of α-actinin concentration, we quantitatively measured the dose-response of Listeria motility to increasing concentrations of α-actinin. Intracellular α-actinin concentration in Listeria-infected cells was adjusted by microinjecting purified concentrations of α-actinin once the Listeria began actively forming their characteristic actin rocket-like tail. The highlight of this work was a measured dose-response curve of Listeria motility to increasing concentrations of α-actinin. Our results show a statistically significant biphasic dependence of Listeria motility to increasing concentrations of α-actinin, with maximal migration rates occurring at an adjusted intracellular concentration predicted $\approx K_D$ of α-actinin (Fig. 5-1). These results underscore the importance

of α-actinin in the intracellular propulsion of Listeria and qualitatively agree with our Actin Filament Zipper models.

In conclusion, in this thesis we have studied two different, but related areas of cell locomotion. These studies are related to two of three steps involved required in tissue cell migration: adhesion and force-generation. The first study was involved in the adhesion, while the second study was involved in understanding how the interaction of two tissue cell cytoplasmic components, namely actin filaments and α-actinin can generate cellular forces not only with regard to the intracellular propulsion of *Listeria monocytogenes*, but equally important in understanding how propulsive forces are generated by tissue cells. For the three-dimensional migration of tissue cells, we found the adhesiveness of the substratum can play a significant and important role in determine how effective the cell is able to migrate. Secondly, we found one could potentially exploit the relationship between adhesion and migration to direct cellular displacements. These studies are particularly important because they represent the first studies to directly observe and statistically relate three-dimensional cell migration as a function of increasing adhesiveness and adhesion gradient steepness. For the second study investigating the interaction of actin filaments and α-actinin, in the context of how this interaction relates potentially to a propulsion mechanism utilized by Listeria, we found through the construction of both thermodynamic and Brownian dynamic models, our models described the dynamic bundling of actin filaments how the chemical energy associated with the interaction could generate significant cellular forces. Lastly, experimentally we tested the motile response of Listeria to increasing concentrations of α-actinin and found the dose-response of Listeria to increasing concentrations of α-actinin to qualitatively agree with our model predictions. Based on both these current experimental evidence and previous experimental evidence (Dold et al., 1994; Sechi et al., 1997; Nanavati et al., 1994) coupled with the current modeling predictions presented here, we feel the actin

filament zipper mechanism plays a significant role in the intracellular propulsion of *Listeria monocytogenes*. We anticipate the zippering mechanism as described here, be it mediated via α-actinin or other molecules, can be used to described a number of other cellular movements and cellular processes such as the intracellular rocketing of vaccinia virus and *Shigella flexneri*, lamellipodia extension and the helical winding of DNA strands following DNA polymerase at the replication fork.

Future Directions

Multi-Filament Actin Network

For the initial modeling efforts describing the actin filament zippering mechanism we focused specifically on the interaction between two actin filaments. However, in actuality, there are a number of filaments involved in the propulsion of Listeria (Fig. 4.1). A next logical advancement to this current work is to extend these models developed here to the case of multi-filament networks. Here, an investigation into actin filament distribution and orientation near more complex-contoured surfaces such as the curved rear of the Listeria bacterium. We expect that in modeling the more complex multi-filament network, our models will give a more accurate estimation of propulsive forces to access the actin filament zipper mechanism and potentially explain how convolved Listeria cell movements, such as continuous circling movements, often seen in the motility of Listeria can occur through perturbations in the actin filament network orientation and/or distribution at the rear of the bacterium.

Vesicle Deformation

Related to the actin-based propulsion of Listeria monocytogenes, recently Taunton et al. used studied spatial and temporal control of actin assembly. They found a subset of the cytoplasmic vesicles enriched in protein kinase C (PKC), caused the vesicles to move through the cytoplasm forming an actin rocket-tail, reminiscent of Listeria motility. The mechanical stiffness of the vesicle membranes is much less than that of a bacterial cell wall as in the case of Listeria. Interestingly, the morphology of the vesicles suggests actin filaments exert forces along the sides of the vesicle consistent with a "squeezing" mechanism rather than a pushing force (Fig 6.1). These observations suggest an actin filament zippering mechanism could be involved in the propulsion of these cytoplasmic vesicles.

Fig 6.1. Cytoplasmic vesicle propelled by actin rocket-like tail. Image taken from (Taunton et al., 2000). Bar scale = 500nm.

Calculating the pressure distribution along the sides of the cytoplasmic vesicles should provide insights into the local pressure exerted by the actin rocket tail and to test if the actin filament zipper model could be used to explain deformed morphologies as observed here. The membrane energy balance can be described by:

$$E_T = \int \left(E_B(\mathbf{r}') + E_0(\mathbf{r})\right) dA \qquad (6.1)$$

where $\mathbf{r}$ is vector position along the membrane coordinate axis, $E_B(\mathbf{r}',\mathbf{r}'')$ accounts for energy associated with deforming the membrane and $E_o(\mathbf{r})$ is a term lumping other energies together including the local energy exerted by actin network.

Minimizing the total energy to of the membrane, we can derive the Euler-Lagrange equations describing the shape of the vesicle membrane:

$$\frac{d^2}{d\mathbf{r}_1^2}\left(\frac{\partial f}{\partial \mathbf{r}_1''}\right)-\frac{d}{dx}\left(\frac{\partial f}{\partial \mathbf{r}_1}\right)+\frac{\partial f}{\partial \mathbf{r}_1}=0 \tag{6.2}$$

$$\frac{d^2}{d\mathbf{r}^2}\left(\frac{\partial f}{\partial \mathbf{r}_2''}\right)-\frac{d}{d\mathbf{r}_2}\left(\frac{\partial f}{\partial \mathbf{r}_2}\right)+\frac{\partial f}{\partial \mathbf{r}_2}=0 \tag{6.3}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the independent coordinate axis and $\mathbf{r}_3 = f(\mathbf{r}_1, \mathbf{r}_2)$. Assuming the equilibrium shapes given in the images and that the only other energy term in addition to the actin-rocket tail contributing to E_o is osmotic energy, then we can solve for $\mathbf{r}_1$ and $\mathbf{r}_2$ component of the actin pressure exerted at a local point (osmotic energy= PV, $P \equiv -RT/\overline{V}_{H20} * \ln(x_{H2O})$, where $\overline{V}_{H20}$ is molar volume of water and x_{H2O} is molar fraction of water in vesicle, V=volume of vesicle). Knowing the components of the pressure exerted by the actin network, we then can easily solve for the magnitude of pressure exerted by the actin network given by:

$$P_{actin} = \sqrt{P_1^2 + P_1^2} \tag{6.4}$$

Measuring the "Stall" Force of *Listeria monocytogenes* Using a Laser Trap

Currently no one to our knowledge has been able to experimentally measure the intracellular propulsive forces involving the pushing the Listeria bacterium primarily because the forces involved pushing the bacterium can be several hundred picoNewtons in magnitude, thus making it difficult to noninvasively measure. One attractive method is to use a three-dimensional optical laser trap. Here, the laser can be focused to a point immediately in the path of the moving Listeria. Once the Listeria migrates into the focused region of the trap, the force exerted by the laser will act to stop and hold the segment of the Listeria focused in the laser trap. In the past we have attempted to do this with a 100mW He-Ne laser trap, however, the laser trapping force proved to be insufficient in effectively stopping the bacteria. That said, the laser trap did have a visible effect many times on the motility of the Listeria. It

is expected, with a more powerful laser trap, one can get an estimate of the forces involved in the propulsion of Listeria. Soon, a variable power 1W laser trap will be on-line in our research group. If the stall force is greater than the forces exerted by the new 1W laser, another approach could be to use high-speed video recording measuring the change in momentum of the bacteria prior to and once within the focused region of the laser trap. Using the experimental data and writing a momentum balance given by:

$$\frac{d}{dt}P_{Tot} = \rho\left\langle v_1^2 \right\rangle - \rho\left\langle v_2^2 \right\rangle - F_{Laser} \tag{6.5}$$

where P_{Tot} ($g \bullet m^{-1} \bullet s^{-1}$) is total momentum ρ ($m \bullet l^{-3}$) is the density of Listeria, v_1 and v_2 is velocity (ms^{-1}) prior to and once within focused region of laser trap, F_{Laser} pressure force ($g \bullet m^{-1} \bullet s^{-1}$) exerted by the trapping laser on the bacteria. The force exerted by the trapping laser could be calibrated by loading micron-sized glass beads following a protocol for loading glass beads into living cells previously described by McNeil and Warder (McNeil and Warder, 1987). Once the beads were inside, one could translate the bead particles inside the cytoplasm of the cell at faster rates until a critical velocity for the particle staying in the trap was obtained. Using that critical velocity one could calculate the drag force on the particle at the critical velocity. The drag force at this velocity would provide an estimate for the trapping force.

APPENDIX A
THREE-DIMENSIONAL CELL TRACKING PROGRAMS

Note: These programs were developed for Optimas™ programming language, Analytical Language for Images.

Bscan3.mac

```
/*Macro designed to recursively scan any number of fields in three-
dimensions, export 3-D positions to Excel file, and update the
position of the particle/cell in the region of interest.

Brian Burgess  Original Version: Feb 1996
                        Current Version:
*/

     freeze();delayms(400);
     selectfullscreen();
     acquire();
     Clearscreen();
     Acquire();
     hLib = LoadMacroLibrary("surface.oml");
     R=2;C=2;   k=c;       //represent row 1 and column 1 Excel
     worksheet 1;
     q=1;j=1;   //represents row 1 and column 1 for sheet 2;
     RunMacro ("C:/burgess/macros/bcomml.mac");
     RunMacro ("C:/burgess/macros/bscini2.mac");
     RunMacro ("C:/burgess/macros/bscinic.mac");

     mwindow(h=0);         // Turn off/on focus window

     cspeed(25000,25000,10000);
     cfspeed(4000);
     show(rfspeed());
     acquire();
     cfstep(100);
/*Number of Cells to be tracked is equal to the number of fields to
scan*/

objective=Prompt("Which Objective Lens are You Using? 20 or 40?"
,"INTEGER");
   if (objective==20)
```

```
    {
       calibrate(cal20x);}
   else
          calibrate(cal40x);
    }

     freeze();
     delayms(400);
     ROIJJ=SelectFullScreen();
     show("The select full screen command holds the values
     ":totext(roijj[0]),"  ":totext(roijj[1]),"  ":totext(roijj[2]),"
     ":totext(roijj[3]));

/* Setting-up 2 Thresholds       */
/* Defining Threshold #1, Set for White Interior Cell     */
     Acquire();
     ty=false;
     while (ty==false)
     {
         show("Move to test cell and hit ok");
         Focus(1); Fwait();
        ty=Prompt(" This Threshold will be set to Black Side; i.e.,
        Does This Cell Exhibit a White Interior?,",2);
     }

     NGrid=6:8;
     Acquire(FALSE);Delayms(2000);
     Th = SetArThresh(NGrid,1);

     Acquire();
/* Defining Threshold #2, Set for Black Interior Cell     */
     ty=false;
     while (ty==false)
      {
      show("Move to test cell and hit ok");
      Focus(1); Fwait();
     ty=Prompt(" This Threshold will be set to White Side; i.e., Does
     This Cell Exhibit a Black Interior?,",2);
      }
     Th2 = SetArThresh(NGrid,1);
     Acquire();

/*Export Documenting Information regarding the Program being executed
to Excel spreadsheet*/
     gotime=getdatetime();
     Message="This is data Stored From Running
     C:/Burgess/Macros/BSCAN3";
     sheet="sheet1";
     hchan= DDEInitiate("Excel",sheet);
     spacing1="R1C4:R1C8";
     spacing2="R1C9:R1C25" ;      // r is the row for the
     different time step; k=cell/column number
```

```
    DDEPoke(hchan, spacing1,gotime);
    DDEPoke(hchan,spacing2,message);
    DDETerminate(hchan);

/* End of Export  */

/* Restart Here if Cell Positioning is Necessary */
     mwindow(0);
     Acquire();
     CELLS=Prompt("How many Cells to Scan", "INTEGER");
     spreadsheetsetup(CELLS);

/* Find the Fields Manually using stage motors */
     show(" Move to reference object");
     Acquire();
     Focus(3);  Fwait();
     freeze();delayms(400);
     Processing2();
     delayms(2000);
     itest=GetShape(MArCentroid);
     if (itest[0]==0)
               { Acquire();
               Show("Pick a NeW Reference Point. This is not
               appropiate.  Hit Enter when Found New Ref.");
                Focus(3);Fwait();
                Freeze();delayms(400);
                Processing2();
                }
     Acquire();
     Here(0,0,0);
     starttime = dostime();                //the first time increment
     will be off3 but thereafter it should be fine
     LONG time=dostime()-starttime;
     delete(message);
     pos=FindingCells(cells);
     //All positions, now increment down a row on worksheet
     R=R+1;k=c;
     mwindow(h=0);

/*All initial positions now are stored in an array*/
/*Stored Initial Positions are also recored in Excel file*/
/* Restart Here if Cell Positioning is Unnecessary */
k=2;

/*Redefine the Clock Timer so Do not Lose Time While Setting Up Cells
in Fields*/
     show(time);time=0;
     starttime=dostime();
     LONG time=dostime()-starttime;
/*Reposition routine*/
     LONG timmax =72000L; show(time,timmax);show(starttime);
     While (time<timmax)
```

```
{
Acquire();delayms(500);
i=0;
      while (i<(CELLS))
            {

           cfstep(100);
            Acquire();
            nx=pos[i,0];
            ny=pos[i,1];
            nf= pos[i,2];
            movabs(nx,ny,nf);     //absolute move
            Wait();    Fwait();//Hold for motors to stop
            freeze();delayms(200);
            chcells=totext(i+1);
            hSObja=TextToScreen ("Cell #", 100.0 :
            175.0,1,0);
            hSObjb = TextToScreen (chcells,140.0 :
            175.0,1,0);
            delayms(400);
            clearscreen(hsobja);
            clearscreen(hSobjb);
            delayms(300);
            Acquire();
            Focus(1);
            Fwait();
            freeze();
            delayms(400);
            Processing();

            itest=GetShape(MArCentroid);
            if (itest[0]==0)
                 {
                 // Applying Processing 2 Function;
                 Acquire();
                 Freeze();delayms(400);
                 Processing2();delayms(400);

                 itest=GetShape(MArCentroid);
                 if (itest[0]>0)
                 {
                   goto CORRECTDISTANCE;
                  }
                   if (itest[0]==0)
                    {

                 Processing 1 For Last Time");
                 Acquire();
                 cfstep(2000);
                 Focus(2);
                 Fwait();
                 cfstep(1000);
```

```
          Focus(3);
          Fwait();
          cfstep(150);
          Freeze();
          delayms(400);
          Processing();
          delayms(400);
            }
            if (itest[0]>0)
          {
            goto CORRECTDISTANCE;
          }

          if (itest[0]==0)
          {
           Message=0;
             goto LOST;
          }
          }

     LABEL CORRECTDISTANCE
      if (itest[0]>0)
          {
          Message=1;
          CreatePoints(mArCentroid);
          ymid=(ROIJJ[1]-ROIJJ[3])/2;
          xmid=(ROIJJ[2]-ROIJJ[0])/2;
          CreatePoints(xymid=xmid:ymid);
Show(xymid);show(mArCentroid);
          Delete(diff);
          Real diff[itest[0],2];
          diff=mArCentroid-xymid;
          dist=diff[,0]*diff[,0]+diff[,1]*diff[,1];
          dmin=min(dist);
          // Return value to spreadsheet as
          dist.traveled;
          nds=GetShape(dist);
          ii=0; lmin=TRUE;
          while (lmin)
                {
                     lmin=(dmin!=dist[ii]);
                     ii=ii+1;
                }
          Acquire();
          LONG mdist[2];
          mdist[0]=1L*(10.*diff[ii-1,0]+.5);
          mdist[1]=1L*(10.*diff[ii-1,1]+.5);
          MovRel(mdist[0],mdist[1],0);
          Wait(); Fwait();
          pos[i, ]=findpos();
//Redefine xyz pos coordinates
          time=dostime() - starttime;
```

```
                }

                LABEL LOST

if ( i==40)          //Loop resets the column for sheet 2 back to 2nd
column for the 42nd cell which is on sheet 2
{
     k=2;
}
if (i==80)
 {
  k=2;     //Loop resets the column for sheet 3 back to 2nd column for
           the 81nd cell which is on sheet 2
  }
coordXYZ(pos,time,message,i,k);  //Write XYZ, time to Excel
k=k+6;
i=i+1;
}
R=R+1;k=c;
}

//Warning Signal Program is Finished, Twinkle Twinkle Little Star!!!
t=0;
while (t<2)
{
Beep(40,4);  Beep(40,4);  Beep(47,4);  Beep(47,4);  Beep(49,4);
Beep(49,4);
Beep(47,2); delayms(200);  Beep(45,4);  Beep(45,4);  Beep(44,4);
Beep(44,4);
Beep(42,4);  Beep(40,2);delayms(100);
t=t+1;
}
```

Bcomm1.mac

```
/* Program sets up communication and inputs all functions to run later
macro*/

DEFINE mwindow(h=0)
/* Function command to turn focus window on/off -- default is on*/
{
          CommOpen ("COM1:9600,N,8,2");
          CHAR box="-";
          if (h==1)
          {box="+";}
          command="Window ":box:" \r";
          CommWrite (0,command);
          DelayMS(100);
          ist=CommStatus(0);
          CommClose(0);
```

```
          return ist;
          }

DEFINE Fspeed(nsp)
/* Function command changing focus motor speed */
{
     nstat=commopen("COM1:9600,N,8,2");
     nstat=0;
     command = "Fspeed  ":totext(nsp):" \r";
     commwrite(nstat,command);
     delayms(200);
     ist=commstatus(nstat);
     Commclose(nstat);
     return ist;
}

DEFINE movabs(nx,ny,nf)
/* Function command for motors -- moves a motor(s) to an absolute
position*/
{
     nstat=commopen("COM1:9600,N,8,2");
     nstat=0;
     command = "move x=":totext(nx):"y =":totext(ny):"f =
     ":totext(nf):" \r";
     commwrite(nstat,command);
     delayms(200);
     ist=commstatus(nstat);
     Commclose(nstat);
     return ist;
}


Define vmove(nx,ny)
/* Function command to move one or two motors simultaneously to an
absolute stage position -- purpose is to run the motors so that they
will plot a straight line, can be used with point ids or motor ids.
Default is set for point ids.*/
{
     nstat=commopen("COM1:9600,N,8,2");
     nstat=0;
     command="vmove x=":totext(nx):"y=":totext(ny):" \r";
     commwrite(nstat,command);
     delayms(200);
     ist=commstatus(nstat);
     commclose(nstat);
     return ist;
}

DEFINE movrel(nx,ny,nf)
/* Function command to move stage to a position relative to present
position. May use point-ids or motor-ids. Default is point-ids.*/
{
```

```
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="movrel x=":totext(nx):"y=":totext(ny):"f =
      ":totext(nf):" \r ";
      commwrite(nstat,command);
      delayms(200);
      ist=commstatus(nstat);
      commclose(nstat);
      return ist;
}

DEFINE Wait()
/* Function waits until all motors stop before continuing in the
program. */
{
          nstat = CommOpen ("COM1:9600,N,8,2");
          nstat=0;
          INTEGER Stest=1:1;
          while (Stest[0]>=0)
          {
               command="status \r";
               CommWrite (0, command);
               DelayMS(200);
               sstring=(CommRead(nstat));
               Stest= SearchString("B",sstring);
          }
          return CommClose(nstat);
}


DEFINE FWait()
/* Function command waits until focus motor stops before continuing.
*/
          {
          nstat = CommOpen ("COM1:9600,N,8,2");
          nstat=0;
          INTEGER Stest=1:1;
          while (Stest[0]>=0)
          {
               command="status F \r";
               CommWrite (0, command);
               DelayMS(200);
               sstring=(CommRead(nstat));
               Stest= SearchString("B",sstring);
          }
          return CommClose(nstat);
          }

DEFINE findpos()
/* Function command returns the current stage position -- can be
specified in point id or motor id form. Default is point id. */
{
```

```
      LONG ips[];
      LONG ivec=0;
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command= "where xyf \r";
      commwrite(nstat,command);
      delayms(200);
      sstring=commread(nstat);
      istat=fromtext(sstring,ips,,3);
      commclose(nstat);
      return ips;
}

DEFINE Here(nx,ny,nf)
/* Function command assigns coordinates to the current position of the
stage --  redefines the coordinates of stage*/
{
      LONG NX,NY,NF;
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="here x=":totext(nx):"Y=":totext(ny):"f=":totext(nf):"
      \r";
      commwrite(nstat,command);
      delayms(200);
      ist=commstatus(nstat);
      commclose(nstat);
      return ist;
}

DEFINE rspeed()
/* Function command to read the motor speed. Command uses motor-ids*/
{
      delete(ta);
      LONG ips[], ta;
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="speed  x y f \r";
      commwrite(nstat, command);
      delayms(300);
      sstring=commread(nstat);
      ta=fromtext(sstring,ips,,3);
      commclose(nstat);
      return ips;
}

DEFINE cspeed(sx,sy,sf)
/* Function command to assign the motor speeds. Command uses motor-ids
*/
{
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
```

```
      command = "speed
      x=":totext(sx):"y=":totext(sy):"f=":totext(sf):"\r";
      commwrite(nstat, command);
      delayms(200);
      ist=commread(nstat);
      commclose(nstat);
      return ist;
}

DEFINE raccel()
/* Function command to read acceleration of the stage-- Ramp time is
inversely proportional to acceleration value (i.e. shorter ramp times
= smaller the acceleration value). Range: 1-255.*/
{
      LONG ips[];
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="accel x y f \r";
      commwrite(nstat,command);
      delayms(300);
      sstring=commread(nstat);
      ta=fromtext(sstring,ips,,3);
      commclose(nstat);
      return ips;
}

DEFINE caccel(ax,ay,af)
/* Function command to change acceleration value. */
{
      LONG AX,AY,AF;
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="accel
      x=":totext(ax):"y=":totext(ay):"f=":totext(af):"\r";
      commwrite(nstat,command);
      delayms(200);
      commclose(nstat);
      return ist;
}


DEFINE homme()
/* Function command to move the specified motors to respective end
limit switches; Once the end limit is activated the position is
zeroed. Default function sends x y and f to end limits*/
{
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command ="home x y f \r";
      commwrite(nstat,command);
      delayms(200);
      ist=commstatus(nstat);
```

```
      commclose(nstat);
      return ist;
}


DEFINE calibration()
/* Function command to center the stage and set position to zero.
Function applies only to x and y. */
{
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="calib s \r";
      commwrite(nstat,command);
      commclose(nstat);
}

DEFINE rfdelay()
/* Function command to read the delay before data reading for
stabilizing focusing motor. Values range: 1-250; each value = 16.6ms;
Default automatically set to 1.*/
{
      LONG ips[1],ta;
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="fdelay \r";
      commwrite(nstat,command);
      delayms(300);
      sstring=commread(nstat);
      ta=fromtext(sstring,ips,,1);

      commclose(nstat);
      return ips;
}

DEFINE cfdelay(fd)
/* Function command to change the framing delay;*/
{
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="fdelay ":totext(fd):" \r";
      commwrite(nstat,command);
      delayms(300);
      commclose(nstat);
}

DEFINE rfspeed()
/* Function command to read focus speed (reported in pps) */
{
      INTEGER ips[1];
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="fspeed \r";
```

```
      commwrite(nstat,command);
      delayms(300);
      sstring=commread(nstat);
      ist=fromtext(sstring,ips,,1);
      commclose(nstat);
      return ips;
}

DEFINE cfspeed(fs)
/* Function command to change focus speed. Nominally set at 20000pps
Values range: 84 and 33000pps*/
{
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="fspeed ":totext(fs):"\r";
      commwrite(nstat,command);
      delayms(200);
      ist=commread(nstat);
      commclose(nstat);
      return ist;
}

DEFINE rfstep()
/* Function command to read the f-step value. Critical in determining
focus distance.   Distance= f-step value*profile dist */
{
      INTEGER ips[1];
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="fstep \r";
      commwrite(nstat,command);
      delayms(200);
      sstring=commread(nstat);
      ist=fromtext(sstring,ips,,1);
      commclose(nstat);
      return ips;
}


DEFINE cfstep(cf)
/* Function command to change the f-step value*/
{
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="fstep ":totext(cf):"\r";
      commwrite(nstat,command);
      delayms(200);
      ist=commread(nstat);
      commclose(nstat);
      return ist;
}
DEFINE rrdprdist()
```

```
/* Function command to read the profile distances that are set for
focusing.*/
{
INTEGER ips[];
     nstat=commopen("COM1:9600,N,8,2");
     nstat=0;
     command="rdprdist \r";
     commwrite(nstat,command);
     delayms(200);
     sstring=commread(nstat);
     ist=fromtext(sstring,ips,,6);
     commclose(nstat);
     return ips;
}


DEFINE cldprdist(chigh,cmed,clow,fhigh,fmed,flow)
/* Function command to change the profile distances set for focusing*/
{
     chigh=200;
     cmed=201;
     clow=210;
     fhigh=20;
     fmed=21;
     flow=23;
     nstat=commopen("COM1:9600,N,8,2");
     nstat=0;
     command="write
     f91=":totext(chigh):"f92=":totext(cmed):"f93=":totext(clow):"f94=
     ":totext(fhigh):"f95=":totext(fmed):"f96=":totext(flow):"\r";
     commwrite(nstat,command);
     delayms(200);
     command2="lprdist \r";
     commwrite(nstat,command2);
     delayms(200);
     show(commread(nstat));
     ist=commread(nstat);
     commclose(nstat);
     return ist;
}

DEFINE focus(f)
/* Function command to focus; Default set to panel mode */
{
     nstat=commopen("COM1:9600,N,8,2");
     nstat=0;
     char fs;fs=1;
       if (f==1)

     //Grey Poupon mustard.
     {  fs="L";
     }
```

```
      if (f==2)
      {  fs="M";
       }
      if (f==3)
      { fs="H";
      }
      command="Focus  ":fs:" \r";
      commwrite(nstat,command);
      delayms(200);
      ist=commread(nstat);
      commclose(nstat);
      return ist;
}

DEFINE signal()
/* Function command to read the focus intensity signal from the
camera. Used for test purposes or fine adjustments. Lower value
represents poor focus and higher value is associated with a clearer
focus.*/
{
      INTEGER ips[];
      nstat=commopen("COM1:9600,N,8,2");
      nstat=0;
      command="sig f \r";
      commwrite(nstat,command);
      delayms(200);
      sstring=commread(nstat);
      ist=fromtext(sstring,ips,,1);
      commclose(nstat);
      return ips;
}
```

Bscinic.mac

```
/*This macro contains functions for Various Programs written and used
by
Brian Burgess  */


hLib = LoadMacroLibrary("surface.oml");

Define Spreadsheetsetup (cells)                          //good
/* this function will set the spreadsheet up for to enter data */
{
      sheet="sheet1";
      sheet2="sheet2";
      sheet3="sheet3";
      hchan= DDEInitiate("Excel",sheet);
      hchan2= DDEInitiate("Excel",sheet2);
      hchan3= DDEInitiate("Excel",sheet3);
```

```
i=1;
k=1;
g=1;        //g is the replacement of k on 2nd sheet
gg=1;       // gg=1 is the replacement of k on 3rd sheet
while (i<(Cells+1))
 {

   if (i <=40)
     {
          j=k+1;
          spacing="R2C":totext(j);
          L=K+2;
          spacing2="R2C":totext(L);
          M=k+2;N=k+3;O=k+4;P=k+5;
          spacing3="R3C":totext(m);
          spacing4="R3C":totext(N);
          spacing5="R3C":totext(O);
          spacing6="R2C":totext(P);
          DDEPoke(hchan,spacing,"Time Field");
          DDEPoke(hchan,spacing2,"Cell  ":
          totext(i):" Position");
          DDEPoke(hchan,spacing6,"Status");
          DDEPoke(hchan,spacing3,"X");
          DDEPoke(hchan,spacing4,"Y");
          DDEPoke(hchan,spacing5,"Z");
          k=k+6;
          i=i+1;
     }

 else if (i>40 && i<=80)
 {
     j=g+1;
     spacing="R2C":totext(j);
     L=g+2;
     spacing2="R2C":totext(L);
     M=g+2;N=g+3;O=g+4;P=g+5;
     spacing3="R3C":totext(m);
     spacing4="R3C":totext(N);
     spacing5="R3C":totext(O);
     spacing6="R2C":totext(P);
     DDEPoke(hchan2,spacing,"Time Field");
     DDEPoke(hchan2,spacing2,"Cell  ":
     totext(i):" Position");
     DDEPoke(hchan2,spacing6,"Status");
     DDEPoke(hchan2,spacing3,"X");
     DDEPoke(hchan2,spacing4,"Y");
     DDEPoke(hchan2,spacing5,"Z");
     g=g+6; i=i+1;
     }
  else
  {
     j=gg+1;
```

```
            spacing="R2C":totext(j);
            L=gg+2;
            spacing2="R2C":totext(L);
            M=gg+2;N=gg+3;O=gg+4;P=gg+5;
            spacing3="R3C":totext(m);
            spacing4="R3C":totext(N);
            spacing5="R3C":totext(O);
            spacing6="R2C":totext(P);
            DDEPoke(hchan3,spacing,"Time Field");
            DDEPoke(hchan3,spacing2,"Cell  ":
            totext(i):" Position");
            DDEPoke(hchan3,spacing6,"Status");
            DDEPoke(hchan3,spacing3,"X");
            DDEPoke(hchan3,spacing4,"Y");
            DDEPoke(hchan3,spacing5,"Z");
            gg=gg+6; i=i+1;
        }
        }
            DDETerminate(hchan);k=c;
            DDETerminate(hchan2);
            DDETerminate(hchan3);
}

DEFINE Spreadsheetfields(fields)
/*Function to label Excel spreadsheet Columns */
{
     sheet="sheet1";
     hchan= DDEInitiate("Excel",sheet);
     i=1;
     k=1;
     while (i<(Fields+1))
       {
           j=k+1;
           spacing="R2C":totext(j);
           DDEPoke(hchan,spacing,"Field ":totext(i));
           k=k+2; i=i+1;
           }
           DDETerminate(hchan);
}

DEFINE Processing()
       {
           freeze();
           Correctbackground(50,50);
           filters(average5x5);
           filters(sharpenhigh);
           filters(invert_binary);
           filters(average5x5);
           filters(average5x5);
           freeze();
           delayms(600);
           Threshold(Th);
```

```
            SetExport(ArCentroid, 1, TRUE);
            SetExport(PtPoints, 1, TRUE);
            AutoExtract = TRUE;delayms(500);
            CreateArea(, FALSE, TRUE);
            delayms(900);
            Return(Arcentroid);
           }

DEFINE Processing2()
        {
            freeze();
            Correctbackground(50,50);
            filters(average5x5);
            filters(sharpenhigh);
            filters(invert_binary);
            filters(average5x5);
            filters(average3x3);
            freeze();
            delayms(2000);
            Threshold(Th2);
            SetExport(ArCentroid, 1, TRUE);
            SetExport(PtPoints, 1, TRUE);
            AutoExtract = TRUE;delayms(500);
            CreateArea(, FALSE, TRUE);
            delayms(2000);
            Return(Arcentroid);
           }

Define CoordXYZ(pos,time,message,i,k)
{
     sheet="sheet1";
     sheet2="sheet2";
     sheet3="sheet3";
     hchan= DDEInitiate("Excel",sheet);
     hchan2= DDEInitiate("Excel",sheet2);
     hchan3= DDEInitiate("Excel",sheet3);
    if (i<=39)
       {
          spacing1="R":totext(r+2):"C":totext(k);
          spacing2="R":totext(r+2):"C":totext(k+1):":R":
          totext(r+2):"C":totext(k+3);
          spacing3="R":totext(r+2):"C":totext(k+4);
          DDEPoke(hchan,spacing1,time);
          DDEPoke(hchan,spacing2,pos[i,]);
          DDEPoke(hchan,spacing3,totext(message));
       }
       else if (i>=40 && i<=79)
          {
          //r is row for the different time step;
          //k=cell/column number
          spacing1="R":totext(r+2):"C":to
          spacing2="R":totext(r+2):"C":totext(k+1):":R":
```

```
            totext(r+2):"C":totext(k+3);
            spacing3="R":totext(r+2):"C":totext(k+4);
            DDEPoke(hchan2,spacing1,time);
            DDEPoke(hchan2,spacing2,pos[i,]);
            DDEPoke(hchan2,spacing3,totext(message));

            }
         else
         {
            spacing1="R":totext(r+2):"C":totext(k);
            spacing2="R":totext(r+2):"C":totext(k+1):":R":
            totext(r+2):"C":totext(k+3);
            spacing3="R":totext(r+2):"C":totext(k+4);
            DDEPoke(hchan3,spacing1,time);
            DDEPoke(hchan3,spacing2,pos[i,]);
            DDEPoke(hchan3,spacing3,totext(message));
         }
            DDETerminate(hchan);
            DDETerminate(hchan2);
            DDETerminate(hchan3);
            return 0;
}

DEFINE TimeStamp(time,k,r)
{
    sheet="sheet1";
    hchan= DDEInitiate("Excel",sheet);
    spacing1="R":totext(R):"C":totext(k);
    // r is the row for the different time step;
    //k=cell/column number
    DDEPoke(hchan,spacing1,time);
    DDETerminate(hchan);
    return 0;
}

Define FindingCells(cells)
{
LONG  pos[CELLS,3];
i=0;
     while (i<(CELLS))
         {

     LABEL topofwhile
     show("Move to CELL Position ":ToText(i+1):
     " Focus Manually,then press any key");
     cfstep(200);
     focus(1);
     Fwait();
     freeze();delayms(400);
     Processing();
     itest=GetShape(MArCentroid);
     if (itest[0]==0)
```

```
        {     Acquire();
              Show("This cell does not fit
              Thresholding Criteria.
              Pick a new Cell.");
              Focus(3);Fwait();
              Freeze();delayms(400);
              Processing();
        }
      Acquire();
      ans=Prompt("Does this Cell Satisfy You
      After Image Processing?",2);
      if (ans==false)
      {
      goto topofwhile;
      }
      show(pos[i, ]=findpos());
      IF (i>0)
           {
           j=0;
           while (j< (i+1))
           {
                if (j!=i)
                  {
                      IF ((ABS(pos[i,0]- (pos[J,0]))) < 800)
                      IF ((ABS(pos[i,1]- (pos[J,1]))) <1000)
                      {

                      Show( "Pick a New Cell - -
                      You Already Used This Cell");
                      goto topofwhile;
                      }
                      }
                      j=j+1;
                      }
                }
                Message=900;
                time=dostime()-starttime;
           if (i==41)
           { k=2;
           }
           if (i==81)
           {
           k=2;
           }
           CoordXYZ(pos,time,message,i,k);
           //writes the postion to the spreadsheet
           i=i+1;k=k+6;
           }
           k=c;
           return pos;
}
```

```
DEFINE FindingFields(Fields,starttime,r)
/*Function to store positions of Fields in an array*/
{
LONG  pos[FIELDS,3];
i=0;k=c;
      while (i<(FIELDS))
          {

      LABEL topofwhile
      show("Move to FIELD Position ":ToText(i+1):
      " Focus Manually,then press any key");
      cfstep(200);
      focus(3);
      Fwait();
      freeze();
      delayms(400);
      Processing();
      itest=GetShape(MArCentroid);
      if (itest[0]==0)
      { Acquire();
            Show("This FIELD does not fit
            Thresholding Criteria.Pick a new Field.");
            Focus(3);Fwait();
            Freeze();delayms(400);
            Processing();
       }
      Acquire();
      ans=Prompt("Does this Field Satisfy You After
      Image Processing?",2);
      if (ans==false)
      {
      goto topofwhile;
      }
      show(pos[i, ]=findpos());
       IF (i>0)
            {
            j=0;
            while (j< (i+1))
            {
            if (j!=i)
               {
                  IF ((ABS(pos[i,0]- (pos[J,0]))) < 1000)
                  IF ((ABS(pos[i,1]- (pos[J,1]))) <1000)
                  {
                  Show( "Pick a New Field - -
                  You Already Used This Field");
                  goto topofwhile;
                  }
                  }
                  j=j+1;
                  }
               }
```

```
          time=dostime()-starttime;
          TimeStamp(time,k,r);
          //writes the position to the spreadsheet
          i=i+1;k=k+2;
          }
               k=c;
               return pos;
          }

DEFINE SetArThresh(NGrid,ResetROI=1)
/*  Function initializes the threshold parameter for identifying area
objects */
          {
               hLib = LoadMacroLibrary("surface.oml");
               ans2=TRUE;
               while(ans2)
               {
                    If (ResetROI==1)
                {
               freeze();Delayms(1000);
               Show("Now Select Your Region of Interest");
               ROI=SelectROI();

                }
               Correctbackground(20,20);
               filters(average5x5);
               filters(sharpenhigh);
               filters(invert_binary);
               filters(average5x5);
               filters(average5x5);

               T=Threshold();
               MinPoints=Prompt("Enter min. number
               pixels/cell","INTEGER");
               REAL RMP;
               RMP=MinPoints;
               AreaCNVFactors[5] =  RMP;
               AreaCNVFactors[7] = TRUE;
               CreateArea (, FALSE, TRUE);
               ans2=Prompt("Try again?",2);
               clearscreen();acquire();
               }
               return T;
          }

Define Recenter(Th,ROIJJ,NGrid)
    {

     Movabs(0,0,0);
     Wait();
     FWait();
     cfstep(35);
```

```
      Focus(3);
      Fwait();
      cfstep(250);
      freeze();
      delayms(400);
     filters(average5x5);
     filters(sharpenhigh);
     filters(invert_binary);
     filters(average5x5);

      Threshold(Th2);
     SetExport(ArCentroid, 1, TRUE);
      SetExport(PtPoints, 1, TRUE);
      AutoExtract = TRUE;delayms(500);
      CreateArea(, FALSE, TRUE);
      ymid=(ROIJJ[1]-ROIJJ[3])/2;
      xmid=(ROIJJ[2]-ROIJJ[0])/2;
      CreatePoints(xymid=xmid:ymid);
     itest=GetShape(MArCentroid);
     if (itest[0]>0)
     {
           Delete(diff);
           Real diff[itest[0],2];
           diff=xymid - mArcentroid;
           dist=diff[,0]*diff[,0]+diff[,1]*diff[,1];
           dmin=min(dist);
           distance=sqrt(dmin);
           // return value to spreadsheet as dist. traveled;
           nds=GetShape(dist);
           ii=0; lmin=TRUE;
           while (lmin)
                 {
                  lmin=(dmin!=dist[ii]);
                   ii=ii+1;
                  }
           Acquire();
           LONG mdist[2];
           mdist[0]=1L*(10.*diff[ii-1,0]+.5);
           mdist[1]=1L*(10.*diff[ii-1,1]+.5);
           MovRel(-mdist[0],-mdist[1],0);
           Wait(); FWait();
           Here(0,0,0);
          }
           Return itest;
         }

DEFINE CountCells(NGrid,Thresh)
/*Funtion counts point objects using a given threshold level */
           {
                 filters(average5x5);
                 filters(sharpenhigh);
                 filters(invert_binary);
```

```
                filters(average5x5);
                filters(average5x5);
                freeze();delayms(600);
                Threshold (Thresh);
                DataCollectionType=0;
                SetExport (mPtSamples, 1, TRUE);
                SetExport (PtPoints, 1, TRUE);
                AutoExtract = TRUE;
                CreatePoints (, FALSE, TRUE);
                if(GetShape(mPtSamples)==0) mPtSamples=0;
                return (mPtSamples);
            }

DEFINE SvImage(i,counter,Fields)
/*Function saves images to Directory preset. Setup for Track 2-D.
Program = 2DIMAGE.MAC    */
{
     if (i<=8)
     {
        type=1;
     }
     if (i>8 && i<=16)
    {
        type = 2;
     }
     if (i>16 && i<=24)
     {
        type = 3;
     }
     if(i>24)
    {
       type =4;
    }

     saveimage("d:/burgess/images2/":totext(type):totext(i):
     "p":totext(counter):".tif");
}
```

APPENDIX B
CELL TRACKING ANALYSIS PROGRAMS

Note: These programs were developed for Matlab™ programming language.

D2cold.m

```
%
% Cell Tracking Algorithm
% Matlab V 5.22 Windows
%
% RB Dickinson, BT Burgess
% Chemical Engineering Department
% University of Florida
% email: dickinso@che.ufl.edu, bburgess@che.ufl.edu
%
%
% This macro loads cell position data and calculates Mean
% Squared Displacement (MSD) as a function of time.
%
%
% "test" is input matrix storing cell track data
% Input data File: Time (sec), X,Y,Z, Status;
% "Status" is an index from cell tracking program indicating
% success of tracking a given cell
%

% Initialization
% Calculate Average Cell Track Time Increment

first          = test(1,:);
sz             = size(first);     % sz Num of Cols (of Input File)
ncells         = sz(2)/5;         % Number of Cells Tracked

col            = test(:,1);
sz             = size(col);       % sz Number of Rows=Number of
                                  % Time Steps
nt             = sz(1);           % Number of tTme Steps
colp           = ([col' 0])';
colp           = colp(2:nt+1);
colp           = colp-col;
tinc           = sum(colp(1:nt-1)./(nt-1));  % tinc= Number of Time Inc
```

```
% Display Values to Command Window
ncells
nt
timeincrement = tinc/60                    % Convert tinc to
                                           % Minutes
% Declare Matrix Size to Store X, Y and Z Displacement
xx            = zeros(nt,ncells);
yy            = xx;
zz            = yy;
nc            = 1;


% Calculate X, Y, Z Distance

for nc=1:ncells            % Load Distance Position Sequences
   xycal      = 0.1;       % Conversion Factor From Motor Steps
                           % to Microns
   zcal       = 0.01;
   xx(:,nc)   = (test(:,(nc-1)*5+2)-test(1,(nc-1)*5+2))*xycal;
   yy(:,nc)   = (test(:,(nc-1)*5+3)-test(1,(nc-1)*5+3))*xycal;
   zz(:,nc)   = (test(:,(nc-1)*5+4)-test(1,(nc-1)*5+4))*zcal;

   %The following plots Total Distance v/s each cell;
   %plot3(xx(:,nc), yy(:,nc),zz(:,nc));     % 3-D Projection of
                                           % Cell Track
   %text(xx(:,nc),'X Distance',yy(:,nc),'Y Distance', zz(:,nc),'Z
    Distance');
    plot(xx(:,nc), yy(:,nc));              % 2-D Projection of
                                           % Cell Track


     hold on;
     axis([-200 200 -200 200]);
     plot(xx(:,nc), yy(:,nc),'*');               % 2-D Projection
                                                 % of  Cell Track
     hold off;
     xlabel('Distance Traveled X (microns)');    % Label Axis
     ylabel('Distance Traveled Y (microns)');
     %zlabel('Distance Traveled Z (microns)');

     pause;                                      % Pause After Each
                                                 % Cell Track Plot
end
% Calculate Squared and Mean-Squared Displacements (MSD)

d2                  = zeros(nt,3);  % Matrix Containing X,Y,Z
                                    % Squared Displacement
d2inc               = zeros(1,3);   % Matrix Containing Mean
                                    % Squared Displacement
ms                  = zeros(nt,ncells);   % Record of Number of
                                          %Continuous Paths
```

```
lim                    = 100^2;              % Max Sq Displmt
                                             % in One Interval
ntr                    = zeros(1,nt);        % Log of Cell Track
                                             % Lengths
for itt                = 1:nt-1              % Itt is Interval
                                             % Size to Calculate
                                             % MSD

  s                    = 0;                  % "s" Counts the
                                             % number of MSD
                                             % intervals summed
  for nc               = 1:ncells

    miss               = 0;                  % "miss" Is Index
                                             % which accounts for
                                             % Cell Track Length

    % Setting Up to Count Distance by Intervals
    % D2inc Vector Contains X,Y,Z MSD
    % It2 is Time Interval Point to Begin to Calculate MSD
    %
    for it2            = 1 : itt : nt-itt
     % Calculate X Squared Dislpacement
     d2inc(1)          = (xx(itt+it2,nc)-xx(it2,nc))^2;

     % Calculate Y Squared Displacement
     d2inc(2)          = (yy(itt+it2,nc)-yy(it2,nc))^2;

     % Calculate Z Squared Displacement
     d2inc(3)          = (zz(itt+it2,nc)-zz(it2,nc))^2;

      if(itt==1)       % For Interval Size=1, If Cell is Lost
                       % (Checking for Large Jumps in Data)

     % Checking if X or Y Squared are Greater than Lim Allowed
     %for an interval = 1
     % If Greater than lim, "miss" is incremented by +1
        if(max(d2inc(1:2))>lim)      % Checking only X,Y
           miss        = miss+1;
        end

        % Mark "miss" value for Cell Path for Each Cell
        ms(it2+1,nc) = miss;
      end

     % Calculate Mean-Squared X,Y,Z for all Continuous Cell
     %Paths

     % Checking if Next Interval Step has the Same "miss" Value
     % If True, then Add the Two Incremented Distances Together
     if(ms(itt+it2,nc)     == ms(it2,nc))
```

```
      d2(itt,:)              =  d2(itt,:)+d2inc;

      % Increment Number of Points to be Averaged
      s                      =  s+1;

       end

     end

   end                              % Increment Next Cell

   % Calculate Average Distance Traveled per Time Interval

   d2(itt,:)              = d2(itt,:)./s;     % MS X,Y,Z for given
                                              % interval size = itt
   % Display Values to Command Window
   [itt s d2(itt,1) d2(itt,2) d2(itt,3)]

end                               % Increment Time Interval

hold off;                         % Allow to Erase Previous Plot

%  Determine Track Lengths for Fit Program
ntr                    = zeros(1,nt);      % Matrix Records
                                           % Length of
                                           % Continuous
                                           % Cell Tracks
for ic                 = 1:nc;

    istep              = 1;                % Counter for
                                           % Continuous Cell
                                           % Track Length
      for itt          = 1 : nt-1;         % Itt = Time Interval
                                           % Size
        if(ms(itt,ic)     ~= ms(itt+1,ic)) % Check if Cell Track
                                           % is Continuous Along
                                           % Interval Length
          ntr(istep)     = ntr(istep)+1;     % Mark Ntr w/ Number
                                             % of Continuous Time
                                             % Intervals
          istep          = 0;                % Reset "istep"

        end

       istep=istep+1;                        % Incrt Track Length

      end

  ntr(istep)=ntr(istep)+1;                   % Record Track
                                             % Continuous Track
                                             % Length for a Cell
```

```
end                                          % Increment to Next
                                             % Cell and Check All
                                             % Intervals
```

Fitexp.m

```
%
% Cell Tracking Algorithm
% Matlab V 5.22 Windows
%
% RB Dickinson, BT Burgess
% Chemical Engineering Department
% University of Florida
% email: dickinso@che.ufl.edu, bburgess@che.ufl.edu
%
%
% This macro Fits Data to Model Using Generalized
% Nonlinear Least-Squares Regression
%
% Output is Optimum Parameter Estimation with Appropriate
% Calculated Standard Deviation Based on Variance-Covariance
% Matrix
%
opt                = 1;              % =1 then MSD based on non
                                     % overlapping intervals, = 0
                                     % then overlapping intervals

ntv                = ntr;            % "ntr" Log of Cont Cell Tracks
par                = [30 1 10];      % "par" initial parameter gues [mu pt
gam]
                                     % mu  = Random Motility
                                     % Coefficent
                                     % pt  = Characteristic
                                     % Persistence Time
                                     % gam = Mean-Squared Positioning
                                     % Error

np                 = 30;             % Number of Data Points to Fit
                                     % to Model
tol                = 1e-8;           % Maximum Error Tolerance
                                     % Allowed for Residuals
% Routine to Fit Data to Model
t                  = 1 : np          % Define Time Interval Vector
dt                 = 1;

[f J]              = eta(np,par); % F = Fit of Model to Data using
                                  % Initial Parameter Guess
                                  % J = Jacobian Matrix of
                                  % Parameters
err2               = 1e4;         % Initialize Error Value > Tol
```

```
% Calculate MSD from Data in D2cold.m
% X^2 + Y^2= d22; For Number of Points to Fit to Model
d22                = d2(1:np,1) + d2(1:np,2);

tp=t*tinc/60;                % Convert "t" Time Interval to Minutes
                             % for Plotting

plot(tp,f,tp,d22);           % Plot Initial Parameter Guess to Data
drawnow;                     % Forces Matlab to Update Plot
pause;


% Iteration Loop to Optimize Model Parameter Estimation

ii                 = 0;
par0               = par;    % "par0" Contains Previous Parameter
                             % Estimations
                             % Iteration Loop to Calculate
                             % Covariance Matrix

while(err2 > tol)

 'Calculating V'
   % "vmatp" Function to Calculate Covariance Matrix
   vp              = vmatp(np,ntv,par,opt,dt)

 err               = 1e10;        % Set Initial Error to Enter
                                  % Iteration Loop

 % Iteration Loop to Fit Parameter Estimates to Model and
 % Calculate Residuals
 % Generalized Nonlinear Least-Squares Regression
 % Seber and Wild (1989)
 %

 while(err > tol)

   u=chol(vp);               % "u" is an Upper Triangle Matrix of
                             % Covaraince Matrix

   [f J]            = eta(np,par)      % "eta" Func Calculates f
                                       % (Model Fit) and J
                                       % (Jacobian)

   res              = d22 - f;         % Residuals = y -f(par)

   kres             = u' \ res;

   K                = u' \ J;
```

```
   % "del" = f(par updated) - f(par old)
   del              = (K'* K ) \ (K' * kres );
   err              = del' * del;

   par              = par + del';      % Update Parameters

   err                                 % Display to Command Window

end

ii                  = ii + 1;          % Iteration Loop Counter

% Determine If Parameter Update is Sufficiently Small
err2                = (par - par0) * (par - par0)';
par0                = par;             % Update Previous Parameter
                                       % Estimation

plot(tp, f, 'r-', tp, d22,'yo');       % Plot Model Fit To Data

hold on;

% Plot Expected Standard Deviation Lines
plot(tp, f+sqrt(diag(vp)),'r:',tp,f-sqrt(diag(vp)),'r:');
xlabel('Time Interval (minutes)');

ylabel('Squared Displacement (um^2)');

drawnow;

hold off;

% Calculation of Variance

p                   = 3;               % "p" is Number of
                                       % Parameters Fit to Model

variance            = (1/( np - p) ) * (res') * (vp^(-1)) * res
                                       %"np" = Num of Observations

% Calculate Varaince-Covaraince Matrix (D)

D                   = variance*inv(K'*K);   % "variance" is a
                                            % scalor; K*K is 3x3

mu                  =[par(1) sqrt(D(1,1))]; % sqrt(D(1,1)) = std
                                            % deviation of mu
                                            % from Diagnol of D

% "Pt" Persistence Time
Pt                  = [par(2) sqrt(D(2,2))];
Gam                 = [par(3) sqrt(D(3,3))]
```

```
% mu= (1/nd)*S^2*Pt: Solved for S = Speed
S(1)                = sqrt(mu(1)*3/Pt(1));
ds(1)               = 3/S(1)/Pt(1)/2;       % d(mu)/dS(1)

ds(2)               = -1/2*S(1)/Pt(1);      % d(pt)/dS(1)

ds(3)               = 0;                    % d(Gam)/dS(1)

S(2)                = sqrt(ds*D*ds');       % Standard Deviation
                                            % of "S"
% Rescale Time Intervals to Minutes

mu                  = mu/tinc*60

Pt                  = Pt*tinc/60

S                   = S/tinc*60

par                             % Display Parameter Values

err2

end
```

Eta.m

```
%
% Cell Tracking Algorithm
% Matlab V 5.22 Windows
%
% RB Dickinson, BT Burgess
% Chemical Engineering Department
% University of Florida
% email: dickinso@che.ufl.edu, bburgess@che.ufl.edu
%
%
% Function Fits Parameter to Model and Calculates
% Jacobian Matrix
%
%
function [f,J] = eta(np,par)

% Define Indices

mu                  = par(1);   % "mu" is Random Motility
                                % Coefficient

p                   = par(2);   % "p" is Persistence Time
```

```
gam                 = par(3);    % "gam" is Mean-Squared
                                 % Positioning Error

t                   = ( 1 : np )'; % "t" is Time Interval

J                   = zeros(np,3);% "J" is Jacobian Matrix of
                                 % mu, p and gam

% f is the Persistent Random Walk Model

f                   = 4 * mu.* ( t - p.* ( 1 - exp( -t./ p ) ) )
                  + 2 * gam;

% J(:,1) = df/dmu
J(:,1)              = 4 * ( t - p.* ( 1 - exp(  -t./ p ) ) );

% J(:,2) = df/dp
J(:,2)              = -4 * mu* ( 1 - ( 1 + t./ p ).* exp( -t./ p  ) );
% J(:,3) = df/dgam
J(:,3)              = 4 * ones(np,1);
```

Omega.m

```
%
% Cell Tracking Algorithm
% Matlab V 5.22 Windows
%
% RB Dickinson, BT Burgess
% Chemical Engineering Department
% University of Florida
% email: dickinso@che.ufl.edu, bburgess@che.ufl.edu
%
% This function is used to determine T1,T2 and T3
% based on ip which is the amount or distance of overlap
% for a given interval. After running this part you will
% have the correlation (w) due to that interval you are at.
% It is returned in the form of omega.
%
%

function om=omegam(i,j,ip,dt,par,ph13a, ph13b, ph2a, ph2b, ph2c)

% Determine T1, T2 and T3 Given the Amount of Overlap for Cell Tracks
Occurring for T2

if(ip >= 0)

  if(ip>=i-j)
```

```
      T1          = ip  +1;
      T2          = (i-ip)  +1;
      T3          = (ip+j-i)  +1;

    end


    if(ip<i-j)

      T1          = ip  +1;
      T2          = j  +1;
      T3          = (i-j-ip)  +1;

    end

    if(ip>=i)

      T1          = 1;
      T2          = 1;
      T3          = 1;

    end

  end

if(ip<0);

    if (ip+j>=i)

      T1          = -ip+1;
      T2          = i+1;
      T3          = (ip+j-i)  +1;

    end

    if (ip+j<i)
      T1          = -ip  +1;
      T3          = (i-j-ip)  +1;
      T2          = (j+ip)  +1;

    end

    if(ip+j<=0)

       T1         = 1;
       T2         = 1;
       T3         = 1;

    end

end
```

```
if(min([T1 T2 T3])<0)

  [i j ip]

  stop

end

[T1 T2 T3];

% Calculate Final Form of Omega (w), given T1,T2, T3 and Phi
% To this point Phi is unweighted for T1,T2,T3

om=ph13a(T1,T3)*ph2a(T2)+ph13b(T1,T2)*ph2b(T2)+ph2c(T2);
```

Phim.m

```
%
% Cell Tracking Algorithm
% Matlab V 5.22 Windows
%
% RB Dickinson, BT Burgess
% Chemical Engineering Department
% University of Florida
% email: dickinso@che.ufl.edu, bburgess@che.ufl.edu
%
%
%
% This function creates all the possible w's or omega's
% Data is stored into the following ph13a,ph2a, etc
% Omegam calculates proper T1,T2,T3
% With T1,T2,T3 then Calculate Correct Form of Omega (om)
%
%
function [ph13a, ph13b, ph2a, ph2b, ph2c] =phim(nt,par,dt)

mu              = par(1);
p               = par(2);

% Time Not Weighted in Calculation  -- T1,T2,T3 are unknown at
% this point
% Here Calculate a Generic Form of Phi (w) and Weight w/ Correct
% Time in Omega.m
%

tm              = 0 : (dt/p) : ( (nt-1) * dt/p );

pr              = mu .* p;
```

```
[tl t3]         = meshgrid(tm);

t2              = tm;

% Integrated Form of phi
ph13a           = 32*pr.^2*(2-exp(-tl)-exp(-t3));

ph2a            = t2.*(1+2.*exp(-t2))-2.*(1-exp(-t2))-.5.*(1-exp(
                  2.*t2));

ph13b           = (1-exp(-tl)).*(1-exp(-t3));

ph2b            = 16.*pr^2.*(1-exp(-2*t2)-2*(t2).*exp(-t2));

ph2c            = 64*pr^2.*(.5*(t2).^2-(t2).*(5/2+2*exp(
                  t2))+.25*(1-exp(2*t2))+4*(1-exp(-t2)));
```

Velem.m

```
%
% Cell Tracking Algorithm
% Matlab V 5.22 Windows
%
% RB Dickinson, BT Burgess
% Chemical Engineering Department
% University of Florida
% email: dickinso@che.ufl.edu, bburgess@che.ufl.edu
%
%
% Function Calculates V  (Full Correlation Matrix) for given "i" and
"j"
%
%

function v=vmat(nt,i,j,dt,par,opt);

%  nt: number of time steps averaged
%  np: number of points to plot
%  par=[mu,p]
%  dt: time increment
%  opt: =1 for non-overlapping incs, =0 for overlapping


[ph13a, ph13b, ph2a, ph2b, ph2c] =phim(nt,par,dt);

% Finding Overlapping Intervals From Cell Track Analysis (T2)

      cnt         = 0;                % Initialize

      su          = 0;
```

```
kk          = l : (l + opt * (i-l) ): nt-i;

szi         = size(kk);

szi         = szi(2);           % n(i) = Number of Possible
                                % "i" Intervals given "nt"

ll          = l : ( l + opt * (j-l) ) : nt-j;

szj         = size(ll);

szj         = szj(2);           % n(j) = Number of Possible
                                % "j" Intervals given "nt"

for k       = kk(1) : i : kk(szi)

% Narrowing down the interval to consider for overlapping
% intervals.
% Determine Amount of Overlap

      test  = find((ll>=max(1,k-j)) & ll<=min(nt-j,i+k));

      for l = ll(min(test) ) : j : ll(max(test) );


         % "ip" = Measure of Overlapping Cell Tracks

         ip     = l - k;

         % Determining the Proper Correlation Based on "ip"
         % and Phi Calculation
         % "omegam" Calculates T1, T2, T3; T2 = Times
         % Corresponding to Overlapping Cell Tracks

         ads    = omegam(i,j,ip,dt,par,phl3a, phl3b, ph2a,
                  ph2b, ph2c);

         % Summing Up the Omegas or Correlation Over a
         % Given Distance
         % "su" = Covariance for a Given "i" and "j"

         su     = su + ads;

  end

end

cnt          = szi * szj;  % Number of Possible intervals
                           % for i and j

% Full Correlation Matrix V=v/nj*ni
```

```
    v            = su/cnt;

    % Accounting For Positioning Error
    if (i==j)

      v=v+8*par(3)^2/(nt-j);

    end
```

Vmatp.m

```
%
% Cell Tracking Algorithm
% Matlab V 5.22 Windows
%
% RB Dickinson, BT Burgess
% Chemical Engineering Department
% University of Florida
% email: dickinso@che.ufl.edu, bburgess@che.ufl.edu
%
%
%
% Function to Calculate Covaraince Matrix
%
function vp       = vmatp(np,ntv,par,opt,dt);    % Call Statement

sz                = size(ntv);                   % "ntv(nt)":
                                                 % Number of
                                                 % Cells Tracked
                                                 % to "nt"
                                                 % Positions

sz                = sz(2);       % "sz(2)" = Number of Time
                                 % Intervals
% Calculation of Weighting Factor (Wm)

wm                = sparse(0);   % Define Wm as Composed Only Nonzero
Elements
for nt            = 2 : sz

  for i           = 1 : (nt-1)

    tmp           = size(1:1+opt*(i-1):nt-i);    % "tmp" =1:1

    wm(i,nt)      = tmp(2);                      % "wm" upper
                                                 % triangle = 1;
                                                 % sparse nonzero
                                                 % elements
```

```
  end

end

% Weighting Factor w(i,:): Equal to Number Cell Tracks for
% Interval "i"/Number of Total Tracks
for i=1:sz-1
    wm(i,:)        = wm(i,:)/sum(wm(i,:)*ntv');

end


% Calculating Pooled Correlation Matrix

vp                 = zeros(np);    % Define Size = Number of Points
                                   % to Fit (np x np)

for i              = 1 : np        % "i" = Interval Size for Cell
                                   % Track Analysis
  i

  for j            = 1 : i         % "j" = Interval Size (j<=i)

    vp(i,j)        = 0;            % Initialize

    % Determine Overlapping Tracks for x(i) and x(j)

    for nt         = i+1 : sz

       if (ntv(nt)> 0 )            % Check Experimental Cell Tracks
                                   % of Length "nt"
                                   % "velem" Accounts For
                                   % Covariance of Overlapping Cell
                                   % Tracks
         vp(i,j)  =
vp(i,j)+ntv(nt)*wm(i,nt)*wm(j,nt)*velem(nt,i,j,dt,par,opt);
% wm is weighting factor

       end

    end

    % "vp" is Lower Triangle Matrix; Symmetric -- v(i,j)=v(j,i)
    vp(j,i)        = vp(i,j);      % "vp" is np x np dimensions

  end

end
```

APPENDIX C
GRADIENT CELL TRACKING ANALYSIS PROGRAMS

Note: These programs were developed for Matlab™ programming language.

Tracksc.m

```
% Program to Compute: Angle Distribution Function; Polar Order
% Parameter; Velocity Track
%
% Brian T. Burgess
% Created 08/12/97
%
format compact
format short

load gd120719.txt;
b=gd120719;
test=[b];

% Initialization
first      = test(1,:);
sz         = size(first);   % size number of columns=no. cells
ncells     = sz(2)/5;
col        = test(:,1);
sz         = size(col);     % size number of rows=number of
                            % time steps
nt         = sz(1);         % no. of time steps
colp       = ([col' 0])';
colp       = colp(2:nt+1);
colp       = colp-col;
tinc=sum(colp(1:nt-1)./(nt-1));  % Calc Time Increment

'NOTE: THIS PROGRAM EXPECTS INPUT DATA TO HAVE IN THE FORM: TIME,Y,X,Z
STATUS'
'X IS THE DIRECTION OF GRADIENT'
ncells
nt
tinc
timeincrement = tinc/60
pause;

% Declare Matrix Size(s)
angledist       = zeros(nt,(nt*ncells));
```

```
velocityx        = angledist;
velocityy        = angledist;
% Velocityx and Velocityy contain the data corr. to velocity
% tracks in x and y dir.

histdx           = angledist;
histdy           = angledist;

% Displacment for all possible intervals for dx and dy

histunitlesscos  = angledist;
histtime         = angledist;
% Trackvcsq contains the data corr. to vel track sq (dx^2 + dy^2)/dt^2
trackvcsq        = angledist;

meandx           = zeros(nt,ncells);
meandy           = meandx;
% These matrices will store all <dx>,<dy> for each interval
% for every cell;

meanrootsqtr     = zeros(nt,1);
meanrootsqvx     = meanrootsqtr;
meanrootsqvy     = meanrootsqtr;
meancos          = meanrootsqtr;
meansin          = meanrootsqtr;

% Set Matrix Size for Histplot -- Stores all Costheta and Sintheta
Measurements for all %Possible  Intervals and all Cells
ci=0;
ty      = 0;

for uuu          = 1:nt

     ty          = ty+uuu;

end

cc               = ty*ncells;
% cc is number of rows for histplot

% Column 1 stores, costheta and Column 2 stores sintheta
histplot         = zeros(cc,2);

rr               = 0;
% rr is row counter for meansqtr,meansqvelx, meansqvely;

% i is the interval size
% Interval Size
% Resets data entry to start on lst column for
% next interval evaluation

for i             = 1 : 3
```

```
    count       = 0;
    col         = 0;      % Resets data to start in col
                          % for each new time interval length

    % Increment to new row for meansqtr,meansqvelx,
    % meansqvely;
    rr          = rr+1;

  % Declare a new Matrix size each time for
  % number of entries in interval matrix;
  % Depends on interval size
  if i          ==0;

    rows        = nt;

  end

  if i          == 1

    rows        = nt-1;

  end

  if i          > 1

      rows      = round(nt/i);

  end

% Declare Temporary Matrices Sizes
  interval          = zeros(rows,ncells);
  unitlesscos       = interval;
  velxsq            = interval;
  velysq            = interval;
  sininterval       = interval;
  trackvelsq        = interval;
  dtime             = interval;
  Polarcos          = interval;
  Polarsin          = interval;
  dx                = interval;
  dy                = interval;
  'interval length is'
  i

% J is the column counter for input file containing  % %
% t,x,y,z,status; J is the increment to the %next cell from %
% input
  for j  = 1:5:(ncells*5 -1)
     % Incrementing by 5 because time,
     % x,y, z, status are given for each cell
```

```
      r           = 1;

      % Resets the row to 1 for each
      % new cell in interval matrix

      col         = col+1;

      % Increment new column for new cell in interval
      % matrix

for k             = 1:(nt-i)

  % Special Case of Interval Size =0;
    if i          == 0
   t0             = test(k,j);     % Units in sec

  y0              = test(k,j+1)*.1;
  % Note: Analyzing gradient; stage labels y as X
  % direction as defined in this program and x as Y
  % direction

   x0             = test(k,j+2)*.1;
  % Note: X and Y units are Converted to Microns
   t1             = test(k+i,j);
   y1             = test(k+i,j+1)*.1;
   x1             = test(k+i,j+2)*.1;
   dt             = (t1-t0)/60;    % Convert dt to minutes
end

t0                = test(k,j);     % Units in sec

y0                = test(k,j+1)*.1;

% Note: Analyzing gradient; stage labels y
% as X direction as defined in this program
% and x as Y direction

x0                = test(k,j+2)*.1;
% Note: X and Y units are Converted to Microns
t1                = test(k+i,j);
y1                = test(k+i,j+1)*.1;
x1                = test(k+i,j+2)*.1;
dt                = (t1-t0)/60;    % Convert dt to min

dxint             = (x1-x0);       % Contains dx over curr
                                   % interval;

dyint             = (y1-y0);       % Contains dy for that
                                   % interval step;


if i              ==0
```

```
% Special Case for Interval =0;
            vxsq     = 0;
            vysq     = 0;
            trvelsq  = 0;

     else

       vxsq          = (abs((x1-x0))/dt)^2;
       vysq          = (abs((y1-y0))/dt)^2;
       trvelsq       = ((y1-y0)^2 + (x1-x0)^2)/(dt^2);

     end

     costheta       = (x1-x0)*(((x1-x0)^2 + (y1-y0)^2)^(
                           .5));     % Units are in Radians

     sintheta       = (y1-y0)*(((x1-x0)^2 + (y1-y0)^2)^(
                           .5

% Convert Cos and Sin Theta to Angles in Units of Radians

    if x1       == x0
      anglecos = acos(0);
      anglesin = asin(1);
    end

    if y1       == y0
      anglecos = acos(1);
      anglesin = asin(0);
    end

    if y1       ~= y0

       if x1    ~= x0

            anglecos       = acos(costheta);
            % Units are in Radians

            anglesin       = asin(sintheta);

       end

     end

    if imag(anglecos) > 0

         'Imaginary Complex Found'
         costheta
         sintheta
         anglecos
         anglesin
```

```
        x1
        x0
        y1
        y0

    end

     %Examine Special Cases for x0,x1 and y0 and y1
     % Special Case #1.
     if x0                  == x1
    %(movement in y direction only)

          % 'Entering Special Case'
          if I              ==0
            ci=ci+1;
          end;

      if y0                 == y1

      anglecos              = 0; % if x0=x1 and y0=y1 then
                                 % cell did not move;

        % Therefore, I will not score
        % Step as a positive or negative step towards the
        % gradient.
        % I will score as 0 for sin and cos theta and for
        % the sin and cos angle
         anglesin  = 0;
         costheta  = 0;
         sintheta  = 0;
      end

     end

     % End of special Case Examination


    if i                ==0
     costheta           = 1;
     sintheta           = 0;
     anglecos           = 0;
     anglesin           = 0;
     end

% Matrix contains all combinations of dx for each cell
%for a given interval length

     dx(r,col)          = dxint;

% Matrix contains all combinations of dy for each cell for a
%given interval length
```

```
    dy(r,col)              = dyint;

    % Interval Matrix contains the Angle in Radians (theta);

    interval(r,col)        = anglecos;
         % Transcribe data to temp matrices;
         % Units in Radians

    unitlesscos(r,col)     = costheta;
         % This is a measure of the collinearity of
         % cell path to the grad: ranges -1 to 1

    % Sintheta Marix contains the Angle in Radians (theta);
    sininterval(r,col)      = anglesin;
    velxsq(r,col)           = vxsq;
    velysq(r,col)           = vysq;
    trackvelsq(r,col)       = trvelsq;
    dtime(r,col)            = dt;

    Polarcos(r,col)          = costheta;
   % Pcos & Psin contain the Cosine and Sine of the Angle;
   % Another Check; Same as w/ Avcos and Avsin
   % Check; I want to be sure I am not transferring
   % data in arrays properly

    if abs(Polarcos)             > 1

         Polarcos
         pause;

    end

    if abs(Polarsin)             > 1

        Polarsin
        pause;

    end

    Polarsin(r,col)             = sintheta;

    r                           = r+1;
     % Increment to the next row for a given cel
  end
    % Stepping through all interval
    % combinations for a given cell

end
    % Increment to next cell

    % Function to Calculate <dx> and <dy> for each interval
```

```
      s                          = size(dx);

      meand                      = zeros(1,s(2));

      for p                      = 1:s(2)

         meand(1,p)              = sum(dx(:,p))/s(1);

      end

      meandx(i+1,:)              = meand;
      meand                      = 0;


      for p                      = 1:s(2)

          meand(1,p)             = sum(dy(:,p))/s(1);

     end

      meandy(i+1,:)              = meand;

      % Determining Matrix Size In order to divide by the
      % correct number of elements (ex. 3x2 = 6 elements)

      tr                       = size(trackvelsq);
      vxsz                     = size(velxsq);
      vysz                     = size(velysq);
      psz                      = size(Polarcos);

% Determining root mean squared velocity and Polar order %Parameter or
average cos, sin for each interval;

% Calculate the mean sq. vel for a given interval size;
meanrootsqtrvel                =
     ((sum(sum(trackvelsq)))/(tr(1)*tr(2)))^(0.5);

meanrootsqvelx                 =
     ((sum(sum(velxsq)))/(vxsz(1)*vxsz(2)))^(0.5);

%Calculate the mean sq. vel for a given interval for vx;

      meanrootsqvely           =
           ((sum(sum(velysq)))/(vysz(1)*vysz(2)))^(0.5);

      avcos                    =
           ((sum(sum(Polarcos)))/(psz(1)*psz(2)));

      avsin                    =
           ((sum(sum(Polarsin)))/(psz(1)*psz(2)));

      meanrootsqtr(rr,1)       = meanrootsqtrvel;
```

```
      meanrootsqvx(rr,l)          = meanrootsqvelx;
      meanrootsqvy(rr,l)          = meanrootsqvely;


% Check to Insure Calculation is OK; If larger than 1 then I will get
an imaginary part for angle

      if abs(avcos)               > 1

            avcos
            pause;

      end


% Check to Insure Calculation is OK; If larger than 1 then I
% will get an imaginary part for angle

      if abs(avsin)               > 1

            avsin
            pause;

     end

      meancos(rr,l)               = avcos;
      meansin(rr,l)               = avsin;

  % Calculation of Matrices for Histogram Plots

  if i                            == 1

       a                          = 1;
       % Sets row cnter for Histplot func to initialize;

  end

  ss                              = size(interval);

  for lp=1:ss(2)

    histplot(a:(a+ss(l)-l),l)     = interval(:,lp)*180/pi;
    % histplot Converted to Degrees
    histplot(a:(a+ss(l)-l),2)     = sininterval(:,lp)*180/pi;
    % Converted to Degrees
    a                                       = a+ss(l);

  end

  % Now Transfer Data in the Interval Matrix to the AngleDist Matrix
for permanent storage;
```

```
  % Angledist is arranged such that each row is increasing interval
size
  %(ex: row 1 = i=1, row2 = i = 2)

  z=size(interval);

  for l=1:z(2)

    move                                        = interval(:,l);
    movevx                                      = velxsq(:,l);
    movevy                                      = velysq(:,l);
    movetrack                                   = trackvelsq(:,l);
    movedx                                      = dx(:,l);
    movedy                                      = dy(:,l);
    switch                                      = unitlesscos(:,l);
    ttime                                       = dtime(:,l);
    angledist(i+1,count+1:count+z(l))           = move';
    velocityx(i+1,count+1:count+z(l))           = movevx';
    velocityy(i+1,count+1:count+z(l))           = movevy';
    trackvcsq(i+1,count+1:count+z(l))           = movetrack';
    histdx(i+1,count+1:count+z(l))              = movedx';
    histdy(i+1,count+1:count+z(l))              = movedy';
    histunitlesscos(i+1,count+1:count+z(l)) = switch';
    histtime(i+1,count+1:count+z(l))            = ttime';
    count                                       = count + z(l);

  end


end   % Increment to New Interval Size


% Calculation of Histograms for Cosine and Sine Theta
% Symmetry for Cosine Theta
% Symmetrycos= -histplot(:,1) + 360;
Symmetrycos                                     = 0;
Symmetrycos                                     = histplot(:,1);

% Symmetry for Sine Theta
% Symmetrysin= -histplot(:,2) + 180;

Symmetrysin                                     = 0;
Symmetrysin                                     = histplot(:,2);
sh                                              = size(histplot(:,1));
ssy                                             = size(Symmetrycos);

% Declare Matrix Size for histcosangle; The only difference %between
histcosangle and
% histplot(:,1) is to account for symmetry;
histcosangle                                    = zeros(cc*2,1);
histsinangle                                    = histcosangle;
histcosangle(1:sh(1),1)                         = histplot(:,1);
```

```
histcosangle((sh(l)+l):(sh(l) + ssy(l)),l)  = Symmetrycos;
histsinangle(l:sh(l),l)                     = histplot(:,2);
histsinangle((sh(l)+l):(sh(l) + ssy(l)),l)  = Symmetrysin;

% Plot Probability of Cosine Angle ;
[N,X]              = hist(histcosangle,30);
sht                = size(histcosangle);
N                  = N/sht(l)*100;
bar(X,N);
ylabel(' Probability of Occurrence (%)');
xlabel(' Angle, Degrees ');
title('Cosine Theta Angle ');

% Plot Probability of Sine Angle;
[N,X]              = hist(histsinangle,50);
sht                = size(histsinangle);
N                  = N/sht(l)*100;
bar(X,N);
ylabel(' Probability of Occurrence (%)');
xlabel(' Angle, Degrees ');
title('Sine Theta Angle ')

'-------------------------------------------------------------'
'Mean Cosine Theta Angle'
s                  = size(histcosangle);
sum(histcosangle)/s(1)
'Stanard Deviation of Cosine Angle'
std(histcosangle)

'Mean Sine Theta Angle'
sum(histsinangle)/s(1)

'Standard Deviation of Sine Angle'
std(histsinangle)

% Calculation of Drift Velocity,V_d
' Mean Drift Velocity is '
ss                 =  size(meanrootsqtr);
sb                 =  size(histplot);

'Size of Histcosangle'
size(histcosangle)


% Base Calculation For Drift on RMS Track Velocity w/ i=1; and
% Mean Costheta for I = 1;
she                =size(histunitlesscos);
drift              =  ( meanrootsqtr(2,l) ) * ( mean(
histunitlesscos(2,1:she(2)-ncells) ) )

' Standard Deviation of V_d is '
std(drift)
```

```
' McCrutheon Index - Length parallel to gradient/ Real Cell Path '
s                = size(meanrootsqtr);
MI               = mean(histunitlesscos(2,:))
plot(meanrootsqtr);
xlabel(' Time Interval ')
ylabel(' Microns / Minute ');
title(' Mean Root Sq Track Velocity ')
'Length Units are in Microns'
' Time Units are in Minutes '
```

klinotxs.m

```
% Program to analyze cell tracks to determine if significant %
% klinotaxis effect
%
% Program developed to segment orientation distribution
% calculated in tracksc.m; Once segmented based on
% orientation w/ respect to gradient data is output
% to 'file' and contains: 'histunitlesscos trackvelocity'
%
%
load file.txt;
ld                       = file;
ss                       = size(ld);
r                        = 0;
% Filter Loaded file to remove all intervals where
% a cell did not displace
%
h=1;
 for i                   = 1:ss(1)

     if ld(i,2)          ~= 0
          r(h,1:2)       = ld(i,1:2);
          h=h+1;
       end
 end

% Convert Orientation Angle w/r/to Gradient to
% Degrees for Ease of manipulation
% New Matrix stored in 'cvtangle'
ss                       = size(r);
cvtangle                 = 0;

for i=1: ss(1)
     cvtangle(i,1)       =acos(r(i,1))*180/pi;
     cvtangle(i,2)       =r(i,2);
end

% Sort and Segment into Discrete Groups According
```

```
% to Orientation w/r/to gradient
%
j=1;k=1;l=1;m=1;n=1;o=1;p=1;q=1;
f15=0;f30=0;f60=0;f90=0;f120=0;f150=0;f165=0;f180=0;

 for i                 = 1:ss(1)

  % +- 15 Segment

  if cvtangle(i,1)>   =-15 & cvtangle(i,1)<=15

 f15(j,1:2)           = cvtangle(i,1:2);
 j                    = j+1;

  % +- 15<x<30 or -15>x>-30
  elseif cvtangle(i,1)>15 & cvtangle(i,1)<=30
  |cvtangle(i,1)<-15 & cvtangle(i,1)>=-30

  f30(k,1:2)           = cvtangle(i,1:2);
  k                    = k+1;

  % +- 30<x<60 or -30>x>-60
  elseif cvtangle(i,1)>30 & cvtangle(i,1)<=60
  |cvtangle(i,1)<-30 & cvtangle(i,1)>=-60

  f60(l,1:2)           = cvtangle(i,1:2);
  l                    = l+1;

  % +- 60<x<90 or -60>x>-90
  elseif cvtangle(i,1)>60 & cvtangle(i,1)<=90 |
  cvtangle(i,1)<-60 & cvtangle(i,1)>=-90

  f90(m,1:2)           = cvtangle(i,1:2);
  m                    =m+1;

  % +- 90<x<120 or -90>x>-120
  elseif cvtangle(i,1)>90 & cvtangle(i,1)<=120 |
  cvtangle(i,1)<-90 & cvtangle(i,1)>=-120

  f120(n,1:2)          = cvtangle(i,1:2);
  n                    = n+1;

  % +- 120<x<150 or -120>x>-150
  elseif cvtangle(i,1)>120 & cvtangle(i,1)<=150 |
  cvtangle(i,1)<-120 & cvtangle(i,1)>=-150

  f150(o,1:2)          =cvtangle(i,1:2);
  o                    = o+1;

% +- 150<x<165 or -150>x>-165
elseif cvtangle(i,1)>150 & cvtangle(i,1)<=165 |
cvtangle(i,1)<-150 & cvtangle(i,1)>=-165
```

```
f165(p,1:2)          = cvtangle(i,1:2);
p                    = p+1;

else
  f180(q,1:2)        = cvtangle(i,1:2);
  q                  = q+1;
end
end


'Mean F15'
a                    = mean(f15)

'Std Error'
s                    = size(f15);
b                    = std(f15)/(sqrt(s(1)))
bb                   = s;

'Mean F30'
c                    = mean(f30)

'Std Error'
s                    = size(f30);
d                    = std(f30)/sqrt(s(1))
dd                   = s;

'Mean F60'
e                    = mean(f60)
'Std Error'
s                    = size(f60);
f                    = std(f60)/sqrt(s(1))
ff                   = s;

'Mean F90'
g                    = mean(f90)
'Std Error'
s                    = size(f90);
h                    = std(f90)/sqrt(s(1))
hh                   = s;

'Mean F120'
i                    = mean(f120)

'Std Error'
s                    = size(f120);
j                    = std(f120)/sqrt(s(1))
jj                   = s;

'Mean F150'
k                    = mean(f150)
```

```
'Std Error'
s                       = size(fl50);
l                       = std(fl50)/sqrt(s(l))
ll                      = s;

'Mean Fl65'
m                       = mean(fl65)
'Std Error'
s                       = size(fl65);
n                       = std(fl65)/sqrt(s(l))
nn                      = s;

'Mean Fl80'
o                       = mean(fl80)
'Std Error'
s                       = size(fl80);
p                       = std(fl80)/sqrt(s(l))
pp                      = s;
z                       = [0 0];
ot          =[a;b;z;c;d;z;e;f;z;g;h;z;i;j;z;k;l;z;m;n;z;o;p;z;];
siz         =[bb;z;z;dd;z;z;ff;z;z;hh;z;z;jj;z;z;ll;z;z;nn;z;z;pp];

wklwrite('out',ot);
```

Orthotxs.m

```
% Program to analyze cell tracks to determine if orthotaxis
% occurred
% Program developed to segment a orientation distribution
% into discrete groups for Orthotaxis
% 'file' should contain 'histunitlesscos trackvelocity'
%
%
load file.txt;
ld                          = file;
ss                          = size(ld);
r                           = 0;
% Filter Loaded file to remove all intervals where
% a cell did not displace
%
h                           = 1;
 for i                      = 1:ss(1)

     if ld(i,2)             ~= 0
          r(h,1:2)          = ld(i,1:2);
          h                 = h+1;
     end

 end
```

```
% Convert Orientation Angle w/r/to Gradient to
% Degrees for Ease of manipulation
% New Matrix stored in 'cvtangle'

ss                          = size(r);
cvtangle                    = 0;
for i                       = 1: ss(1)
     cvtangle(i,1)          = acos(r(i,1))*180/pi;
     cvtangle(i,2)          = r(i,2);
end


% Sort and Segment into Discrete Groups According
% to Orientation w/r/to gradient
%
j=1;k=1;l=1;m=1;n=1;o=1;p=1;q=1;
fl5=0;f30=0;f60=0;f90=0;fl20=0;fl50=0;fl65=0;f180=0;

 for i                      = 1:ss(1)

  % +- 15 Segment
  if cvtangle(i,1)>=-15 & cvtangle(i,1)<=15
     fl5(j,1:2)             = cvtangle(i,1:2);
     j                      = j+1;

  % +- 15<x<30 or -15>x>-30
  elseif cvtangle(i,1)>15 & cvtangle(i,1)<=30 |
  cvtangle(i,1)<-15 & cvtangle(i,1)>=-30
     f30(k,1:2)             = cvtangle(i,1:2);
     k                      = k+1;


  % +- 30<x<60 or -30>x>-60
  elseif cvtangle(i,1)>30 & cvtangle(i,1)<=60 |
  cvtangle(i,1)<-30 & cvtangle(i,1)>=-60
     f60(l,1:2)             = cvtangle(i,1:2);
     l                      = l+1;

  % +- 60<x<90 or -60>x>-90
  elseif cvtangle(i,1)>60 & cvtangle(i,1)<=90 |
  cvtangle(i,1)<-60 & cvtangle(i,1)>=-90
     f90(m,1:2)             = cvtangle(i,1:2);
     m                      = m+1;


   % +- 90<x<120 or -90>x>-120
   elseif cvtangle(i,1)>90 & cvtangle(i,1)<=120 |
   cvtangle(i,1)<-90 & cvtangle(i,1)>=-120
     fl20(n,1:2)            = cvtangle(i,1:2);
     n                      = n+1;
```

```
   % +- 120<x<150 or -120>x>-150
   elseif cvtangle(i,1)>120 & cvtangle(i,1)<=150 |
   cvtangle(i,1)<-120 & cvtangle(i,1)>=-150
     f150(o,1:2)           = cvtangle(i,1:2);
     o                     = o+1;



   % +- 150<x<165 or -150>x>-165
   elseif cvtangle(i,1)>150 & cvtangle(i,1)<=165 |
   cvtangle(i,1)<-150 & cvtangle(i,1)>=-165
     f165(p,1:2)           = cvtangle(i,1:2);
     p                     = p+1;


   else
      f180(q,1:2)          = cvtangle(i,1:2);
      q                    = q+1;

   end
end

'Mean F15'
a                          = mean(f15)
'Std Error'
s                          = size(f15);
b                          = std(f15)/(sqrt(s(1)))
bb                         = s;
'Mean F30'
c                          = mean(f30)
'Std Error'
s                          = size(f30);
d                          = std(f30)/sqrt(s(1))
dd                         = s;
'Mean F60'
e                          = mean(f60)
'Std Error'
s                          = size(f60);
f                          = std(f60)/sqrt(s(1))
ff                         = s;

'Mean F90'
g                          = mean(f90)
'Std Error'
s                          = size(f90);
h                          = std(f90)/sqrt(s(1))
hh                         = s;

'Mean F120'
I                          = mean(f120)
'Std Error'
s                          = size(f120);
```

```
j                               = std(f120)/sqrt(s(1))
jj                              = s;

'Mean F150'
k                               = mean(f150)
'Std Error'
s                               = size(f150);
l                               = std(f150)/sqrt(s(1))
ll                              = s;


'Mean F165'
m                               = mean(f165)
'Std Error'
s                               = size(f165);
n                               = std(f165)/sqrt(s(1))
nn                              = s;

'Mean F180'
o                               = mean(f180)
'Std Error'
s                               = size(f180);
p                               = std(f180)/sqrt(s(1))
pp                              = s;

z                               = [0 0];

ot              =[a;b;z;c;d;z;e;f;z;g;h;z;i;j;z;k;l;z;m;n;z;o;p;z;];
siz         =[bb;z;z;dd;z;z;ff;z;z;hh;z;z;jj;z;z;ll;z;z;nn;z;z;pp];

wklwrite('out',ot);
```

Fuldxdy.m

```
% Program should be used after tracksc.m
% Purpose of the program is to sort out any "jumps"
% in the tracking program.  Large jumps can be associated
% with program loosing a cell during the tracking experiment.
%
% histdx and histdy should be loaded.
%
%
% Brian T. Burgess     05-27-98
%
% Load file Containing Dx (Row 1) and Dy (Row 2) for i=1 (Units should
be in Microns);

% Set Max Cell Velocity
Maxvelocity     = 1000          % mircons/min;
```

```
Dxav              = 0;
Dyav              = 0;
Stddevx           = 0;
Stddevy           = 0;
Samples           = 0;
Meanstanderx      = 0;
Meanstandery      = 0;
Driftx            = dxav;
Drifty            = dxav;
jay               = 1;
cleardx           = 0;
cleardy           = 0;
ss                = size(histdx);
% Calculation of Drift Velocity for interval size=1;
cumdrift          =     histdx(2,1:(nt*ncells
                  ncells))./histtime(2,1:(nt*ncells-ncells));

% Calculation of Standard Drift Vel
stddrift          = std(cumdrift)

for i                   = 2: ss(1)  % Step through interval data

  Timeincrement         = (tinc/60)*(i-1);
  Maxdisp               = Timeincrement*Maxvelocity;
  cleardx               = 0;
  cleardy               = 0;
  jay                   = 1;

  for j                 = 1: (ss(2) - (i-1)*ncells)

      if abs(histdx(i,j)) < Maxdisp &  abs(histdy(i,j)) <
     Maxdisp

     cleardx(jay,1)     = histdx(i,j);
     cleardy(jay,1)     = histdy(i,j);
       jay              = jay + 1;

      end

  end

   a                                = size(cleardx);
   samples((i-1),1)                 = a(1);
   stddevx((i-1),1)                 = std(cleardx(:,1));
   stddevy((i-1),1)                 = std(cleardy(:,1));
   meanstanderrorx((i-1),1)         = stddevx((i-1),1)/(sqrt(samples((i-
1),1)));
   meanstanderrory((i-1),1)         = stddevy((i-1),1)/(sqrt(samples((i-
1),1)));
   stddevy((i-1),1)                 = std(cleardy(:,1));
   dxav((i-1),1)                    = mean(cleardx);
   dyav((i-1),1)                    = mean(cleardy);
```

```
   driftx((i-1),1)          = mean(cleardx)/Timeincrement;
   drifty((i-1),1)          = dyav((i-1),1)/Timeincrement;
   tim((i-1),1)             = Timeincrement;

end
% Calculate Statistical Significance
'Confidence Interval for alpha=.1 (90%)'
'Student t-Distribution, Normal Distribution '

conflev                     = input('Input Significance Value
                              to Evaluate (Alpha set up for a=.05
                              or .1)   ');

% Degrees of Freedom
df              = samples(1,1)-1;

if (conflev     == .1)
   if df>       = 19
    if df       == 19
      t         = 1.328
    end
    if df       ==20
      t         =1.325
    end
    if df       == 21
      t         = 1.323
    end
    if df       ==22
      t         = 1.321
    end
    if df       == 23
      t         = 1.319
    end
    if df       == 24
      t         = 1.318
    end
    if df       == 25
      t         = 1.316
    end
    if df       == 26
      t         = 1.315
    end
    if df       == 27
      t         = 1.314
    end
    if df       == 28
      t         = 1.313
    end
    if df       == 29
      t         = 1.311
    end
    if df       >= 30
```

```
      t          = 1.282
    end

  else

   'No Confidence Interval Calculated'

  end

end

if conflev       == .05
   if df         >= 19
    if df        == 19
      t          = 1.729
    end

    if df        == 20
      t          = 1.72
    end
    if df        == 21
      t          = 1.721
    end

    if df        == 22
      t          = 1.717
    end

    if df        == 23
      t          = 1.714
    end

    if df        == 24
      t          = 1.711
    end

    if df        == 25
      t          = 1.708
    end

    if df        ==26
      t          =1.706
    end

    if df        == 27
      t          = 1.703
    end

    if df        == 28
      t          = 1.701
    end
```

```
    if df        == 29

      t          = 1.699
    end

    if df        >= 30
      t          = 1.645
    end
  else

   'No Confidence Interval Calculated'

  end
 end

% Student-t Distribution Formula
if df            >= 19

   xinterval     = t*stddevx(1,1)/sqrt(samples(1,1));
   yinterval     = t*stddevy(1,1)/sqrt(samples(1,1));
   driftinterval = t*stddrift/sqrt(samples(1,1));

end

'-------------------------------------------------'

'Alpha Value'
conflev

'Confidence Interval for X data'
xinterval

'Mean Displacement in X Dir for Interval Size = 1 '
dxav(1,1)

'Confidence Interval for Y data'
yinterval

'Mean Displacement in Y Dir for Interval Size = 1 '
dyav(1,1)

'Mean Standard Error in X Direction'
meanstanderrorx(1,1)

'Mean Standard Error in Y Direction'
meanstanderrory(1,1)

'Mean Drift Velocity (microns/min) '
mean(cumdrift)

'Confidence Interval for Drift Vel in X dir. (+- microns/min) '
driftinterval
```

```
'Standard Error in the Mean for Drift Vel. (+- microns/min) '
stddrift/sqrt(samples(1,1))

'Mean Root Sq Vel X Dirction'
de              = meanrootsqvx(1:2,1);

'Standard Error Mean Root Sq Vel X Direction'
sr              = size(velocityx);
sty             = sqrt(velocityx(2,:));
sx              = std(sty)/sqrt(sr(2));
dq              = [de(1,1) sx]

'Mean Root Sq Vel Y Direction'
da              = meanrootsqvy(1:2,1);

'Standard Error Mean Root Sq Vel Y Direction'
sr              = size(velocityy);
sty             = sqrt(velocityy(2,:));
sy              = std(sty)/sqrt(sr(2));

dy              = [da(1,1) sy]
```

Turnan.m

```
% Program calculates the Turn Angle w/r/t adjacent
% successive time intervals.
%
% Brian Burgess      Created on: 12-5-97
%
'If cell does not move between Time Increment, Then Angle Labeled as -
900'

format compact
format long

% Load Input File
load txgbr62.txt;
b=txgbr62(1:20,:);
sz          = size(b);
% Number of Time Points
Nt          = sz(1);

% Determine Number of Cells
first       = test(1,:);
sz          = size(first);   % number of columns=no. cells
ncells      = sz(2)/5;
col         = test(:,1);
sz          = size(col);     % rows=number of time steps
```

```
nt          = sz(1);          % no. of time steps

% Number of Turn Angles to be calculated
NumTurnAngle= nt-2;


% Declare Turnangle Matrix Size
Angles      = zeros((nt-2),ncells);

C           = 0;              % Dummie variable to keep up w/
                              % current cell #

% Setup loop to increment cell by cell

for k       = 1:5:ncells*5

   c        = c+1;

   for j    = 1:(nt-2)        % nt-2 is the Number of Turn Angles

     x1     = test(j,(k+1));      % j is the Row; k+1 is column
     x2     = test((j+1),(k+1));
     x3     = test((j+2),(k+1));

     y1     = test(j,(k+2));      % loading up y positions
     y2     = test((j+1),(k+2));
     y3     = test((j+2),(k+2));

     % Calculate dx and dy for each vector r1 and r2
     dx1    = x2-x1;
     dy1    = y2-y1;

     dx2    = x3-x2;
     dy2    = y3-y2;

     if dx1      ~=0 & dy1~=0
        if  dx2  ~=0 & dy2~=0

     % Calculate Dot Product
     Dotprod     = dx1*dx2 + dy1*dy2;

     % Calculate Magnitude of r1 and r2
       mag1           = (dx1^2 + dy1^2)^.5;


       mag2           = (dx2^2 + dy2^2)^.5;

     % Calculation of cosine Angle
       cosineangle=(Dotprod)/(mag1*mag2);
       turnangle= acos(cosineangle)*180/pi;

          else
```

```
            turnangle       = -900;

          end

       else

          turnangle    = -900;

       end

       % Store Turnangle Value
       Angles(j,c)     = turnangle;

     end
end

sz                     = size(Angles);
z                      = 0;
for i                  = 1:sz(1)

  for j                = 1:sz(2)

     if Angles(i,j)    ~= -900

       z               = z+1;
       FilterAngles(z,1)     = Angles(i,j);

     end

   end

end

j                      = 1;
for i                  = 1:ncells

  ang(j:j+(nt-2)-1,1) = Angles(1:nt-2,i);
  ang(j+nt-2,1)       = -1000;
  j                   = j+nt-2+1;

end
```

APPENDIX D
COMPUTER CODE FOR THERMODYNAMIC ACTIN FILAMENT ZIPPER MODEL

Note: These programs were developed to be run using MATLAB software.

Masinh.m

```
% Master program to Inhibitor Species Program
clear;                          % Clear all local variables
pars            = 0;
pars(1)         = 10;           % rho = R_T/Kd2(y=0);
pars(2)         = .5;           % a = A/Kd1;
pars(3)         = 5;            % epsilon = L^2*R_T*KT/B
pars(4)         = 20;           % alpha = L/sqrt(KT/gamma)
pars(5)         = 0;            % inhibitory protein: I/Kd1
rr              = 0;

for power2      = 1:1

     rho        = 10.^power2
     pars(1)    = rho;
     ctlp       = 0;
     rr         = rr+1;

     for L      = 250 :100:2500

          L
          %a                = 10.^power
          %pars(2)          =a;
          %pars(5)          =g;

          ctlp              = ctlp+1;       % loop counter

          inhbbend;                     % Run program inhbbend.m

          % Store Forces
          F1(rr,ctlp)       = Fapp;
          F2(rr,ctlp)       = Fexact;
          F3(rr,ctlp)       = Fchm;
          F4(rr,ctlp)       = Ftot;

   end
```

```
end

% Output results to text file
fid                    = fopen('Gl0.txt','w');
fprintf(fid,'%12.8f    ',F3);
fclose(fid);
fid                        = fopen('Glb0.txt','w');
fprintf(fid,'%12.8f    ',F4);
fclose(fid);
```

Inhbbend.m

```
kT    = 4.1;            % Thermal Energy
P     = 10000;          % Persistent length
%L    = 500;            % Length of actin filament
B     = kT*P;           % Bending modulus
Rt    = 1/40;           % 1 actinin site per 35nm = 1/35 nm^-1

N     = 21;             % Number of increments along x,y axis

xe    =.99;             % Xend; X-coordinate at end of filament
dx    = xe/(n-1);       % incremented x-distance
x     = 0:dx:xe;        % X vector
y0    = 0*x;            % Initial guess for filament shape
y(n)  = .3;


inhbend;                    % Run inhbend.m

% Rescale For Calculation of Curvature
y     = y*L;
dx    = dx*L;

% Exact Method for Calculating curvature
for i       = 2:n-1

     df2   = (y(i+1) -2*y(i) + y(i-1))/dx^2;      % d2f/dx2
     df1   = (y(i+1)-y(i-1))/(2*dx);
     k(i)  = ((df2/(1+df1*df1)^1.5)^2)*dx;

end

k2          = sum(k);
Emech       =.5*k2;


% Approx Method for Calc curvature
sum         = 0;
for i       = 2:n-1
```

```
  sum       = summ+((y(i+1)-2*y(i)+y(i-1))^2/dx^4)*dx;

end
Em          = summ/2;              %Emech=Emech*L/B;
Echem       = kT*Rt*sum(log(1-uu1-uu2-uu3)*dx);
plot(y);
drawnow; hold;

%%% Make a Small increment in Filament
y           = y/L;                 % Scale "y" to dimensionless form
dxe         = .005;
xe2         = xe+dxe;
dx          = xe2/(n-1);
x           = 0:dx:xe2;

inhbend;                           % Run inhbend.m

% Rescale For Calculation of Curvature
y           = y*L;
dx          = dx*L;
dxe         = dxe*L;

% Exact Method for Bending Energy
for i       = 2:n-1
     df2    = (y(i+1) -2*y(i) + y(i-1))/dx^2;      % d2f/dx2
     df1    = (y(i+1)-y(i-1))/(2*dx);
   k(i)     = ((df2/(1+df1*df1)^1.5)^2)*dx;

end

k2          = sum(k);
Emech2      = .5*k2;


% Approx for Bending Energy
sum         = 0;
for i       = 2:n-1

  sum       = summ+((y(i+1)-2*y(i)+y(i-1))^2/dx^4)*dx;

end

Em2         = summ/2;  %Emech=Emech*L/B;

plot(y);
drawnow;hold;
clf;

Echem2      = kT*Rt*sum(log(1-uu1-uu2-uu3)*dx);
```

```
Fexact      = -(Emech2-Emech)/(dxe);
Fapp        = -(Em2-Em)/(dxe);
Fch         = -(Echem2-Echem)/(dxe);

% Force Due to Mechanical Energy
Fapp        = Fapp*B;
Fexact      = Fexact*B;
Fchm        = Fch;
Ftot        = Fexact+Fchm


'End'
```

Inhbend.m

```
% inhBEND.M   - BTB  1/9/00

%  This program calculates the shape of a filament, y(x) of fixed
%  length, L, bound at (0,0) and (1,ye).

%  This program requires functions FBEND and FUNC in the same
%  directory.

%  It is currently set to solve d^4y/dx^4 + alpha*y = 0, which
%  corresponds to a parabolic interaction energy between the
%  filament at the line at x=0.

%  The routine determines the necessary slope, lam=dy/dx at
%  (1,ye), that yields the correct length, L.

%  Note convergence is not guaranteed.

% Initial guess for slope - from previous bbend.m program
lam   = 1;
del   = 0*x;

% Run parameters
err2  = 1e6;
tol2  = 1e-5;
dlam  = 1e-8;

while (err2>tol2)
     % Use Newton's method to find root
     % Calculates function set to zero
     [ff,y,uu1,uu2,uu3,rrho,fvecinh,J]=inhfbend(lam,y0,dx,pars);
     [ff2,y2,uu1,uu2,uu3,rrho,fvec2inh,J]=inhfbend(lam+dlam,y0,
     x,pars);                          % Same, at increment in lam

     dellam     = -ff*dlam/(ff2-ff);  % Variation in lam
```

```
      lam         = lam+dellam;          % Update
      err2        = dellam^2+ff^2;       % Test convergence
      y0          = y;                   % Store previous y
end
```

Inhfbend.m

```
% Function to Minimize Euler-Lagrange Equation function
[diff,y,uu1,uu2,uu3,rrho,f,J] = inhfbend(lam,y0,dx,pars)

y     = y0;
n     = length(y);
err   = 1e6;
tol   = 1e-5;

while (err>tol);

  % Use Newton's method to find root
  [f,J,uu1,uu2,uu3,rrho]    = inhfuncv(y,dx,lam,pars);
  del                       = J\(-f');
  y                         = y+del';
  err                       = del'*del+f*f';

end

% Calculate length of filament
Ltry                        = sum(sqrt(dx^2+(y(1:n-1)-y(2:n)).^2));

% Calculate deviation from true (dimensionless) length
Diff                        = Ltry-1;
```

Funcv.m

```
% Function calculates the Euler-Lagrange function and its
% Jacobian solving f_i(y)=0; J is df_i/dy_j


function [f,J,uu1,uu2,uu3,rrho]=inhfuncv(y,dx,lambda,pars,f1)

%Dimensionless Parameter values
rho0        = pars(1);
a           = pars(2);
epsilon     = pars(3);
alpha       = pars(4);
g           = pars(5);

% Initialize all vectors
```

```
minn        = 1e-6;
f           = 0*y;
n           = length(y);
J           = zeros(n,n);
uu1         = f;
uu2         = f;
uu3         = f;
rrho        = f;

   f(1)     = y(1);     % Forcing Slope =0
   J(1,1)   = 1;        % Derivative is constant
   %f(2)    = y(2);     % Forcing Slope =0
   %J(2,2)  = 1;
   x        = 0:dx:1;


i=1;

      rho         = rho0*exp(-alpha^2*y(i)^2);
      drhody      = -2*alpha^2*y(i)*rho;
      eta         = sqrt(g^2+2*g+2*a*g+1+2*a+a^2+4*rho*a);
      uu1(i)      = 1/2/rho*(-g-1-a+eta);
      uu2(i)      = (1-uu1(i) )/( 1+1/rho/uu1(i)*(1+g) );
      uu3(i)      = g*( 1-uu1(i)-uu2(i) )/(1+g);
      rrho(i)     = rho;

i=2;
     rho          = rho0*exp(-alpha^2*y(i)^2);
     drhody       = -2*alpha^2*y(i)*rho;
     eta          = sqrt(g^2+2*g+2*a*g+1+2*a+a^2+4*rho*a);
     if g         == 0  % For very small values of rho
           u3     = 0;
           u2     = 0;
           u1     = a/(1+a);
     else
           u3     = 1/(1+(1+a)/g);
           u1     = a*(1-u3)/(1+a);
           u2     = 0;

     end

     if(rho>minn)
           uu1(i)= 1/2/rho*(-g-1-a+eta);
           uu2(i)= (1-uu1(i) )/( 1+1/rho/uu1(i)*(1+g) );
           uu3(i)= g*( 1-uu1(i)-uu2(i) )/(1+g);
     end

          rrho(i)=rho;

% Make curvature at i=1 =0; Linear Slope
% At i=2; 4th order derivative as before, except y(0) which is
% i=1-1 is needed;
```

```
% From Finite Diff formula for Curvature=0: y(0)=y(i+1)

ytemp0      = 2*y(1)-y(2);

%Finite Diff approx. for 4th order derivative
f(i)        =y(i+2)-4*y(i+1)+6*y(i)-4*y(i-1)+ytemp0;

% Jacobian
J(i,i+2)    = 1;
J(i,i+1)    =-4;
J(i,i)      = 6;
J(i,i-1)    = -4;
J(i,i-2)    = 1;

% Add Chemical Energy
rho         = rho0*exp(-alpha^2*y(i)^2);

if g        == 0            % For very small values of rho
      u3    = 0;
      u2    = 0;
      u1    = a/(1+a);
else
      u3    = 1/(1+(1+a)/g);
      u1    = a*(1-u3)/(1+a);
      u2    = 0;
end


if(rho>minn)
      drhody      = -2*alpha^2*y(i)*rho;              %dp/dy
      eta         = sqrt(g^2+2*g+2*a*g+1+2*a+a^2+4*rho*a);
      u1          = 1/2/rho*(-g-1-a+eta);
      u2          = (1-u1 )/( 1+1/rho/u1*(1+g) );
      u3          = g*( 1-u1-u2 )/(1+g);

      du1drho     = -1/2/rho^2*(-g-1-a+eta) +a/rho/eta;

      cet         = 1+1/rho/u1*(1+g);
      dadrho      = (1+g)*(1/u1/rho^2+du1drho/rho/u1^2)/cet/cet;
      dtdrho      = -du1drho/cet-u1*dadrho;

      du2drho     = dadrho+dtdrho;
      du3drho     = -g/(1+g)*(du1drho+du2drho);


      % d(u1+u2+u3)/dp:= dG/dp
      b           = du1drho+du2drho+du3drho;

      b           = -b;                 % derivative is
                                        % d(u1+u2+u3)/dp;
```

```
      h           = b*drhody;              % dG/dp*dp/dy:= dG/dy
                                  % G:=h==dG/dp*dp/dy
      correct     = 1/(1-u1-u2-u3);        % correction for Ln term
                                           % from derivative
      f(i)        = f(i)+epsilon*correct*h*dx^4;

      dzdrho      = (-g-1-a+eta)/rho^3-a/eta/rho^2;
      dcdrho      = -a/eta/rho^2 - 2*a^2/rho/eta^3;

      d2u1drho    = dzdrho+dcdrho;

      phi         = 1/u1/rho^2+1/rho/u1^2*du1drho;

      d2adrho     = (1+g)/cet/cet*(-2/u1/rho^3
2/rho/rho/u1^2*du1drho2/rho/u1^3*(du1drho)^2+1/rho/u1^2*d2u1drho+2*(ph
i)*(1+g)*phi/cet);

      d2tdrho     = -d2u1drho/cet-du1drho*(1+g)*(phi)/cet/cet-
du1drho*dadrho-u1*d2adrho;

      d2u2drho    = d2adrho+d2tdrho;

      d2u3drho    = -g/(1+g)*(d2u1drho+d2u2drho);

      dbdy        = d2u1drho+d2u2drho+d2u3drho;
      dbdy        = -dbdy;    % derivative is really
                              % d2(u1+u2+u3)/dp;
      dhdy        = dbdy*drhody;

      d2rhody     = 2*alpha^2*rho*(-1+2*alpha^2*y(i)^2);

      % Working on 1/(1-u1-u2-u3) term
      dbetady     = drhody*(du1drho+du2drho+du3drho)/(1-u1-u2-u3)^2;

      J(i,i)      = J(i,i)+epsilon*(b*drhody*dbetady +
correct*drhody*dhdy + correct*b*d2rhody)*dx^4;

     end

      uu1(i)      = u1;
      uu2(i)      = u2;
      uu3(i)      = u3;
      rrho(i)     = rho;


      for i       = 3:n-2

       %Finite Diff approx. for 4th order derivative
      f(i)        = y(i+2)-4*y(i+1)+6*y(i)-4*y(i-1)+y(i-2);
```

```
% Jacobian
      J(i,i+2)    = 1;
      J(i,i+1)    = -4;
      J(i,i)      = 6;
      J(i,i-1)    = -4;
      J(i,i-2)    = 1;

      % Add Chemical Energy
      rho         = rho0*exp(-alpha^2*y(i)^2);

      if g        ==0   % For very small values of rho
            u3    = 0;
            u2    = 0;
            u1    = a/(1+a);
      else
            u3    = 1/(1+(1+a)/g);
            u1    = a*(1-u3)/(1+a);
            u2    = 0;
      end

      if(rho>minn)

            drhody      = -2*alpha^2*y(i)*rho;              %dp/dy
            eta         = sqrt(g^2+2*g+2*a*g+1+2*a+a^2+4*rho*a);
            u1          = 1/2/rho*(-g-1-a+eta);
            u2          = (1-u1 )/( 1+1/rho/u1*(1+g) );
            u3          = g*( 1-u1-u2 )/(1+g);

            du1drho     = -1/2/rho^2*(-g-1-a+eta) +a/rho/eta;

            cet         =  1+1/rho/u1*(1+g);
            dadrho      = (1+g)*(1/u1/rho^2
                          +du1drho/rho/u1^2)/cet/cet;

            dtdrho      = -du1drho/cet-u1*dadrho;

            du2drho     = dadrho+dtdrho;


            du3drho     = -g/(1+g)*(du1drho+du2drho);


            % d(u1+u2+u3)/dp:= dG/dp
            b           =du1drho+du2drho+du3drho;

            % derivative is really -d(u1+u2+u3)/dp;
            b           = -b;

            % dG/dp*dp/dy:= dG/dy
            % G:=h==dG/dp*dp/dy

            h           = b*drhody;
```

```
          % correction for Ln term from derivative
          correct     = 1/(1-u1-u2-u3);
          f(i)        = f(i)+epsilon*correct*h*dx^4;

          dzdrho      = (-g-1-a+eta)/rho^3-a/eta/rho^2;
          dcdrho      = -a/eta/rho^2 - 2*a^2/rho/eta^3;

          d2uldrho    = dzdrho+dcdrho;

          phi         = 1/u1/rho^2+1/rho/u1^2*duldrho;

          d2adrho     = (1+g)/cet/cet*(-2/u1/rho^3-
2/rho/rho/u1^2*duldrho-
2/rho/u1^3*(duldrho)^2+1/rho/u1^2*d2uldrho+2*(phi)*(1+g)*phi/cet);

          d2tdrho     = -d2uldrho/cet-duldrho*(1+g)*(phi)/cet/cet-
duldrho*dadrho-u1*d2adrho;

          d2u2drho    = d2adrho+d2tdrho;


          d2u3drho    = -g/(1+g)*(d2uldrho+d2u2drho);

          % derivative is really -d2(u1+u2+u3)/dp;
          dbdy        = d2uldrho+d2u2drho+d2u3drho;
          dbdy        = -dbdy;
          dhdy        = dbdy*drhody;

          d2rhody     =2*alpha^2*rho*(-1+2*alpha^2*y(i)^2);

          % Working on 1/(1-u1-u2-u3) term

          dbetady     = drhody*(duldrho+du2drho+du3drho)/(1-u1-u2-
u3)^2;


          J(i,i)      = J(i,i)+epsilon*(b*drhody*dbetady +
correct*drhody*dhdy + correct*b*d2rhody)*dx^4;

          end

          uu1(i)      = u1;
          uu2(i)      = u2;
          uu3(i)      = u3;
          rrho(i)     = rho;

   end


          ytemp       = y(n-1)+2*lambda*dx;
          f(n-1)      = ytemp-4*y(n)+6*y(n-1)-4*y(n-2)+y(n-3);
```

```
          J(n-1,n-3) = 1;
          J(n-1,n-2) = -4;
          J(n-1,n-1) = 7;
          J(n-1,n)   = -4;

% Add Chemical Energy
i                    = n-1;
     rho   =rho0*exp(-alpha^2*y(i)^2);

     if g  ==0        % For very small values of rho

          u3    = 0;
          u2    = 0;
          u1    = a/(1+a);

     else
          u3    = 1/(1+(1+a)/g);
          u1    = a*(1-u3)/(1+a);
          u2    = 0;

     end

     if(rho>minn)

     drhody     = -2*alpha^2*y(i)*rho;      %dp/dy
     eta        = sqrt(g^2+2*g+2*a*g+1+2*a+a^2+4*rho*a);
     u1         = 1/2/rho*(-g-1-a+eta);
     u2         = (1-u1 )/( 1+1/rho/u1*(1+g) );
     u3         = g*( 1-u1-u2 )/(1+g);

     du1drho    = -1/2/rho^2*(-g-1-a+eta) +a/rho/eta;
     cet        =  1+1/rho/u1*(1+g);
     dadrho     = (1+g)*(1/u1/rho^2+du1drho/rho/u1^2)/cet/cet;
     dtdrho     = -du1drho/cet-u1*dadrho;
     du2drho    = dadrho+dtdrho;
     du3drho    = -g/(1+g)*(du1drho+du2drho);

     % d(u1+u2+u3)/dp:= dG/dp
     b          = du1drho+du2drho+du3drho;
     b          = -b; % derivative -d(u1+u2+u3)/dp;

     % dG/dp*dp/dy:= dG/dy
     % G:=h==dG/dp*dp/dy
     h          = b*drhody;

     % correction for Ln term from derivative
     correct    = 1/(1-u1-u2-u3);


     f(i)       = f(i)+epsilon*correct*h*dx^4;
```

```
      dzdrho      = (-g-1-a+eta)/rho^3-a/eta/rho^2;
      dcdrho      = -a/eta/rho^2 - 2*a^2/rho/eta^3;
      d2uldrho    = dzdrho+dcdrho;
      phi         = 1/u1/rho^2+1/rho/u1^2*duldrho;
      d2adrho     = (1+g)/cet/cet*(-2/u1/rho^3-2/rho/rho/u1^2*duldrho-
2/rho/u1^3*(duldrho)^2+1/rho/u1^2*d2uldrho+2*(phi)*(1+g)*phi/cet);

      d2tdrho     = -d2uldrho/cet-duldrho*(1+g)*(phi)/cet/cet-
duldrho*dadrho-u1*d2adrho;
      d2u2drho    = d2adrho+d2tdrho;
      d2u3drho    = -g/(1+g)*(d2uldrho+d2u2drho);
      dbdy        = d2uldrho+d2u2drho+d2u3drho;
      % derivative is -d2(u1+u2+u3)/dp;
      dbdy        = -dbdy;
      dhdy        = dbdy*drhody;
      d2rhody     = 2*alpha^2*rho*(-1+2*alpha^2*y(i)^2);

      % Working on 1/(1-u1-u2-u3) term
      dbetady     = drhody*(duldrho+du2drho+du3drho)/(1-u1-u2-u3)^2;

      J(i,i)      = J(i,i)+epsilon*(b*drhody*dbetady +
      correct*drhody*dhdy + correct*b*d2rhody)*dx^4;
       end

      uu1(i)      = u1;
      uu2(I)      = u2;
      uu3(i)      = u3;
      rrho(i)     = rho;

      % Using finite diff formula for second order derivative
      % set =0; Thus curvature at y(n)=0;
      f(n)        = ytemp-2*y(n)+y(n-1);

      J(n,n)      = -2;
      J(n,n-1)    = 2;
   i              = n;

        rho       = rho0*exp(-alpha^2*y(i)^2);
        drhody    = -2*alpha^2*y(i)*rho;


       if g       ==0   % For very small values of rho
           u3     = 0;
           u2     = 0;
           u1     = a/(1+a);
       else

           u3     = 1/(1+(1+a)/g);
           u1     = a*(1-u3)/(1+a);
           u2     = 0;

       end
```

```
if(rho>minn)

   eta   = sqrt(g^2+2*g+2*a*g+1+2*a+a^2+4*rho*a);
   u1    = 1/2/rho*(-g-1-a+eta);
   u2    = (1-u1 )/( 1+1/rho/u1*(1+g) );
   u3    = g*( 1-u1-u2 )/(1+g);

end

uu1(i)   = u1;
uu2(i)   = u2;
uu3(i)   = u3;


rrho(i)  = rho;
```

APPENDIX E
COMPUTER CODE FOR BROWNIAN DYNAMICS ACTIN FILAMENT ZIPPER MODEL

Bsimple12.m

```
% Program to Numerically Simulate Spring-Bead Model of
% Actin Filament Polymer
% Based on Rouse Model
%
%
%
%
% Forces Accounted for:
% Bead Spring; Bending; Chemical;
% Repulsion (Repulsion modeled as Linear power)
%
% Filaments ARE fixed at Distal End.
%
% BTB 7-18-2000
%
clear;
clf;

%%% Parameters
% 1:= Chemical Interaction Included, 2: Not Included
chem.      = 1;
nd         = 3;       % Number of Dimensions
nfmt       = 2;       % Number of filaments to simulate;
nbeads     = 11;
rmin       = 1;       % rmin/L := Dim Optimal separation dist
                      % between beads
rmin2      = .5;      % Dimensionless Optimal separation dist for
                      % X-Link Bonds: r/a
a          = 100;     % Dimensional spacing between Beads (nm)
kT         = 4.1;     % Thermal Energy pN/nm
dt         = 1e-4;    % rmin^2/D := Dimen. Time increment
nt         = 1.5e6;   % Number of Time Steps
Bm         = 100;     % Dimensionless Bending Modulus: lp/a
Gam        = 300      % Dimensionless Spring Bead Stiffness:
                      % Gam(spring)a^2/KT
Gam2       = 300;     % Dimensionless X-Link Bond Stiffness:
                      % Gam(x-link)a^2/KT
```

```
% a/(KT/Gam2)^.5
alpha       = 1/sqrt(4.1/Gam2);


A           = 1*(7e-7);       % Concentration of Actinin (M)3.5e-7
Kf          = 6.8e5;          % X-Linking Binding Affinity: Units=
                              % (M*s)^-1
Kr          = .4;             % X-Linking Disassociation Constant:
                              % Units= s^-1

m           = 1/(kT/a^2);     % Dimensionless Repulsion Spring:
                              % df/dz
incp        = 5000;
nplot       = incp;

%%% Initialize
r           = zeros(nbeads*nfmt,nd);    % r holds positions of all
                                        % beads;
f           = r;                        % f holds summed forces on
                                        % each bead
fxlink      = r;                        % fxlink contains X-linking
                                        % forces on beads
state       = zeros(nbeads*nfmt,1);     % records Bead Reaction
                                        % State(0,1,2)
link        = zeros(nbeads*nfmt,1);     % Records filmt num & bead
                                        % label that is X-Linked w/
                                        % that Bead
FWALL       = [];
P           = 0;
StWall      = 0;

% Set State Vector for nbeads to nbeads-1; [No chemical
% interaction

for ii      = 1:2

   state(nbeads*(ii-1)+nbeads-1,1)     = -1;
   state(nbeads*(ii-1)+nbeads,1)       = -1;

end


% Chemical Interaction Not Included
if chem.          ==2

  state(:,1)     = -1;

end


%%% Generate Initial Distribution of Filament
%%% Assume Filaments Sep by Distance 'a'
```

```
r =[     0    1.1320         0
    0.9900    0.8900         0
    1.9800    0.6630         0
    2.9700    0.4620         0
    3.9600    0.2980         0
    4.9500    0.1740         0
    5.9400    0.0880         0
    6.9300    0.0360         0
    7.9200    0.0100         0
    8.9100         0         0
    9.9000         0         0
         0   -1.1320         0
    0.9900   -0.8900         0
    1.9800   -0.6630         0
    2.9700   -0.4620         0
    3.9600   -0.2980         0
    4.9500   -0.1740         0
    5.9400   -0.0880         0
    6.9300   -0.0360         0
    7.9200   -0.0100         0
    8.9100         0         0
    9.9000         0         0];

r(:,2)     = r(:,2)+.5;

%%% Plot Routine
for I      = 1:nfmt
   if nd   == 2
      plot(r(nbeads*(i-1)+1:nbeads*(i-1)+nbeads,1),r(nbeads*(I
1)+1:nbeads*(i-1)+nbeads,2),'ro');
      hold on; drawnow;

      plot(r(nbeads*(i-1)+1:nbeads*(i-1)+nbeads,1),r(nbeads*(i-
1)+1:nbeads*(i-1)+nbeads,2),'k-');

      hold on; drawnow; axis([-2 8 -5 5])
      wx    = [0 0];
      wx2   = wx-.1;
      wy    =[-5 5];
      plot(wx,wy,'b-',wx2,wy,'w-');    % Draw Wall
      hold on; drawnow;

      else

      plot3(r(nbeads*(i-1)+1:nbeads*(i-1)+nbeads,1),r(nbeads*(i-
1)+1:nbeads*(i-1)+nbeads,2),r(nbeads*(i-1)+1:nbeads*(i-
1)+nbeads,3),'ro');hold on;
      drawnow;

      plot3(r(nbeads*(i-1)+1:nbeads*(i-1)+nbeads,1),r(nbeads*(i-
1)+1:nbeads*(i-1)+nbeads,2),r(nbeads*(i-1)+1:nbeads*(i-
1)+nbeads,3),'k-');hold on;
```

```
      drawnow;
      axis([-2 8 -5 5 -5 5])
      az    = 8;
      el    = 15;
      view(az,el);xlabel('X');ylabel('Y');
      wx    = [0 0];
      wx2   = wx-.1;
      wy    = [0 0];
      wz    = [-5 5];
      plot3(wx,wy,wz,'b-',wx2,wy,wz,'b-');
      hold on; drawnow; % Draw Wall

   end
end

clf;
rhold      =r;        % Store Initial position of filament

%%% Simulate Brownian Motion
for it     = 1:nt

 for ic    = 1:nfmt

    vps    = 1:nd;

    %%% Nth Bead
    ip     = nbeads;              % Tethered Bead at end

      %%% Spring Force
      dist = r(nbeads*(ic-1)+ip-1,vps)-r(nbeads*(ic-1)+ip,vps);
      dmag = sqrt(sum(dist.^2));
      fmag = Gam*(dmag-rmin);
      fspl = dist/dmag*fmag;

      %%% Bending Force
      if nd      ==2

        beadn    = r(nbeads*(ic-1)+ip,vps)+[rmin 0];
        beadnn   = r(nbeads*(ic-1)+ip,vps)+[2*rmin 0];

      else

        beadn    = r(nbeads*(ic-1)+ip,vps)+[rmin 0 0];
        beadnn=r(nbeads*(ic-1)+ip,vps)+[2*rmin 0 0];

      end

      curve      = r(nbeads*(ic-1)+ip-2,vps)-2*r(nbeads*(ic-1)+ip-
1,vps)+r(nbeads*(ic-1)+ip,vps);

      fbend      = -Bm*curve;
```

```
%%% Chemical Force

[link,state,fxlink]=chem(ip,nbeads,ic,vps,state,link,fxlink
,A,Kf,Kr,dt,alpha,Gam2,nfmt,nd,r,rmin2);


%%% Wall Force
dist      = r(nbeads*(ic-1)+ip,vps)-0;

if dist(1)<0

     if nd==2

       dist2    = [dist(1) 0];

     else

        dist2   = [dist(1) 0 0];

     end

        fmagw   = -m*dist2(1);

       if nd    ==2

         fwall  = [fmagw 0];

        % Store Wall Force
        StWall  = StWall+fmagw;

      Else

        Fwall   = [fmagw 0 0];
        StWall  = StWall+fmagw;

     end


     else

     if nd      == 2

        fwall   = [0 0];

     else

        fwall   =[0 0 0];

     end
```

```
        end


            f(nbeads*(ic-1)+ip,vps)=fsp1+fbend+fwall
+fxlink(nbeads*(ic-1)+ip,vps);

      clear dist dist2 curve dmag fsp1 fwall fmagw

      %%% N-1 Bead
      ip    = nbeads-1;

      %%% Spring Force
      dist  = r(nbeads*(ic-1)+ip-1,vps)-r(nbeads*(ic-1)+ip,vps);
      dmag  = sqrt(sum(dist.^2));
      fmag  = Gam*(dmag-rmin);
      fsp1  = dist/dmag*fmag;

      dist2 = r(nbeads*(ic-1)+ip+1,vps)-r(nbeads*(ic-1)+ip,vps);
      dmag2 = sqrt(sum(dist2.^2));
      fmag2 = Gam*(dmag2-rmin);
      fsp2  = dist2/dmag2*fmag2;

      %%% Bending Force
      curve = r(nbeads*(ic-1)+ip-2,vps)-4*r(nbeads*(ic-1)+ip-
1,vps)+5*r(nbeads*(ic-1)+ip,vps)-2*r(nbeads*(ic-1)+ip+1,vps);

      fbend = -Bm*curve;

      %%% Chemical Force
[link,state,fxlink]=chem(ip,nbeads,ic,vps,state,link,fxlink,A,Kf,
Kr,dt,alpha,Gam2,nfmt,nd,r,rmin2);

     %%% Wall Force
      dist            = r(nbeads*(ic-1)+ip,vps)-0;

      if dist(1)<0

           if nd      == 2

                dist2= [dist(1) 0];

           else

                dist2 = [dist(1) 0 0];

           end

                fmagw = -m*dist2(1);

             if nd    == 2

                fwall =[fmagw 0];
```

```
            % Store Wall Force
            StWall=StWall+fmagw;

         Else

            Fwall   = [fmagw 0 0];
            StWall  = StWall+fmagw;

         end

      else

         if nd      == 2

            fwall   = [0 0];

         else

            fwall   = [0 0 0];

         end

      end

         f(nbeads*(ic-1)+ip,vps)=fspl+fsp2+
fbend+fwall+fxlink(nbeads*(ic-1)+ip,vps);

   clear dist dist2 curve dmag dmag2 fspl fsp2 fbend fwall
fmagw

  for ip   = nbeads-2:-1:3

    %%% Spring Force
      dist = r(nbeads*(ic-1)+ip-1,vps)-r(nbeads*(ic-1)+ip,vps);
      dmag = sqrt(sum(dist.^2));
      fmag = Gam*(dmag-rmin);
      fspl = dist/dmag*fmag;

      dist2= r(nbeads*(ic-1)+ip+1,vps)-r(nbeads*(ic-1)+ip,vps);
      dmag2= sqrt(sum(dist2.^2));
      fmag2= Gam*(dmag2-rmin);
      fsp2 = dist2/dmag2*fmag2;

      %%% Bending Force
      curve= r(nbeads*(ic-1)+ip-2,vps)-4*r(nbeads*(ic-1)+ip-
1,vps)+6*r(nbeads*(ic-1)+ip,vps)-4*r(nbeads*(ic-
1)+ip+1,vps)+r(nbeads*(ic-1)+ip+2,vps);

      fbend= -Bm*curve;
```

```
      %%% Chemical Force
[link,state,fxlink]=chem(ip,nbeads,ic,vps,state,link,fxlink,A,Kf,
Kr,dt,alpha,Gam2,nfmt,nd,r,rmin2);

      %%% Wall Force
      dist = r(nbeads*(ic-1)+ip,vps)-0;

      if dist(1)<0
           if nd      ==2

             dist2    = [dist(1) 0];

           else

              dist2   = [dist(1) 0 0];

           end
              fmagw   = -m*dist2(1);

             if nd    ==2

               fwall  =[fmagw 0];

              % Store Wall Force
              StWall  =StWall+fmagw;
            else

              fwall   = [fmagw 0 0];
              StWall  = StWall+fmagw;

           end

        else

           if nd      ==2

              fwall   = [0 0];

           else

              fwall   = [0 0 0];

           end

        end

      f(nbeads*(ic-1)+ip,vps)=fsp1+fsp2+fbend+fwall+
fxlink(nbeads*(ic-1)+ip,vps);

      clear dist dist2 curve dmag dmag2 fsp1 fsp2 fbend fwall
fmagw
```

```
   end

    %%% i              = 2;
    ip                 = 2;

      %%% Spring Force
      dist             = r(nbeads*(ic-1)+ip-1,vps)-r(nbeads*(ic-
1)+ip,vps);

      dmag             = sqrt(sum(dist.^2));
      fmag             = Gam*(dmag-rmin);
      fsp1             = dist/dmag*fmag;

      dist2            = r(nbeads*(ic-1)+ip+1,vps)-r(nbeads*(ic
1)+ip,vps);

      dmag2            = sqrt(sum(dist2.^2));
      fmag2            = Gam*(dmag2-rmin);
      fsp2             = dist2/dmag2*fmag2;


      %%% Bending Force
      curve            = -2*r(nbeads*(ic-1)+ip-1,vps)+5*
r(nbeads*(ic-1)+ip,vps)-4*r(nbeads*(ic-1)+ip+1,vps)+r(nbeads*(ic-
1)+ip+2,vps);

      fbend            = -Bm*curve;


      %%% Chemical Interaction
[link,state,fxlink]=chem(ip,nbeads,ic,vps,state,link,fxlink,A,Kf,
Kr,dt,alpha,Gam2,nfmt,nd,r,rmin2);

      %%% Wall Force
      dist             = r(nbeads*(ic-1)+ip,vps)-0;

      if dist(1)<0
           if nd       ==2

             dist2     =[dist(1) 0];

           else

              dist2    =[dist(1) 0 0];

           end

              fmagw    = -m*dist2(1);

             if nd     ==2

                fwall       =[fmagw 0];
```

```
            % Store Wall Force
            StWall      = StWall+fmagw;

         else

            fwall       = [fmagw 0 0];
            StWall      = StWall+fmagw;

         end

      else

         if nd          == 2

            fwall       = [0 0];

         else

            fwall       = [0 0 0];

         end

      end


   f(nbeads*(ic-1)+ip,vps)=fspl+fsp2+fbend
+fwall+fxlink(nbeads*(ic-1)+ip,vps);


   clear dist dist2 curve dmag dmag2 fspl fsp2 fbend fwall
fmagw


   %% i            = 1 (Free-End Bead)
   ip              = 1;

   %%% Spring Force
   dist2           = r(nbeads*(ic-1)+ip+1,vps)-r(nbeads*(ic-
1)+ip,vps);

   dmag2           = sqrt(sum(dist2.^2));
   fmag2           = Gam*(dmag2-rmin);
   fsp2            = dist2/dmag2*fmag2;

   %%% Bending Force
   curve           = r(nbeads*(ic-1)+ip,vps)-2*r(nbeads*(ic-
1)+ip+1,vps)+r(nbeads*(ic-1)+ip+2,vps);

   fbend           = -Bm*curve;


   %%% Chemical Interaction
```

```
    [link,state,fxlink]=chem(ip,nbeads,ic,vps,state,link,fxlink
,A,Kf,Kr,dt,alpha,Gam2,nfmt,nd,r,rmin2);

    %%% Wall Force
    dist            = r(nbeads*(ic-1)+ip,vps)-0;

    if dist(1)<0
        if nd       == 2

          dist2     = [dist(1) 0];

        else

           dist2    = [dist(1) 0 0];

        end
             fmagw = -m*dist2(1);

          if nd     == 2

             fwall = [fmagw 0];

             % Store Wall Force
             StWall= StWall+fmagw;

         else

           fwall    = [fmagw 0 0];
           StWall   = StWall+fmagw;

        end

      else
        if nd       == 2

           fwall    = [0 0];

        else

           fwall    = [0 0 0];

        end

      end


    f(nbeads*(ic-1)+ip,vps)=fsp2+fbend+fwall+fxlink(nbeads*(ic-
1)+ip,vps);

    clear dist2 dmag2 fmag2 fsp2 curve fbend fmagw

    %%% Weiner Process
```

```
%%% 2nd Order Approximation for Weiner Process
%%% dW= 1/6 probability +-sqrt(3dt) and 2/3 prob. =0

prb              = rand(nbeads-2,nd);

for k            = 1:nbeads-2;

   if prb(k,1)   <= 2/3

     prb(k,1)    = 0;

   elseif prb(k,1)    <= (2/3+1/6)

      prb(k,1)        = 1;

   else

      prb(k,1)        = -1;

   end

   if prb(k,2)        <= 2/3

      prb(k,2)        = 0;

   elseif prb(k,2)    <=(2/3+1/6)

      prb(k,2)        = 1;

   else

      prb(k,2)        = -1;

   end

   if nd              == 3

      if prb(k,3)     <= 2/3

       prb(k,3)       = 0;

      elseif prb(k,3) <= (2/3+1/6)

       prb(k,3)       = 1;

      else

       prb(k,3)       = -1;

     end

   end
```

```
    end

    % Increment in Weiner Process dW
    dW                      = sqrt(3*dt)*prb;

    % Full Weiner Process
    %dW                     = sqrt(dt*rmin^2)*randn(nbeads-2,nd);

    dr                      = f(nbeads*(ic-1)+1:nbeads*(ic-
1)+nbeads-2,vps)*dt+dW;


   %%% Update Positions
   r(nbeads*(ic-1)+1:nbeads*(ic-1)+nbeads-2,vps)=r(nbeads*(ic-
1)+1:nbeads*(ic-1)+nbeads-2,vps)+dr;

    end

    %%%% Store Wall Forces
    p           = p+1;
  if p          == 1

      %Store Average of 1000 increments of Force
      q         = (StWall)*(kT/a);
      FWALL     = [FWALL q];
      StWall    = 0;
    clear q;p= 0;

  end

  %%% Write Data to File
   if it        == nplot

       fid      = fopen('test.txt','a');

   end

     fprintf(fid,'%12.6f\n',FWALL);
     fclose(fid);
     mean(FWALL)
     FWALL      =     [];
   end


    % Plot Filament Dist every 'nplot' iterations
    if(it==nplot)
    for ij      = 1:nfmt
    if nd       == 2
    plot(r(nbeads*(i-1)+1:nbeads*(i-1)+nbeads,1),r(nbeads*(i-
1)+1:nbeads*(i-1)+nbeads,2),'ro');hold on; drawnow;
```

```
      plot(r(nbeads*(i-1)+1:nbeads*(i-1)+nbeads,1),r(nbeads*(i-
1)+1:nbeads*(i-1)+nbeads,2),'k-');hold on; drawnow;

      axis([-2 8 -5 5])
      wx    = [0 0];
      wx2   = wx-.1;
      wy    = [-5 5];
      plot(wx,wy,'b-',wx2,wy,'w-');hold on;
      drawnow; % Draw Wall

   else

      plot3(r(nbeads*(i-1)+1:nbeads*(i-1)+nbeads,1),r(nbeads*(i-
1)+1:nbeads*(i-1)+nbeads,2),r(nbeads*(i-1)+1:nbeads*(i-
1)+nbeads,3),'ro');hold on; drawnow;

      plot3(r(nbeads*(i-1)+1:nbeads*(i-1)+nbeads,1),r(nbeads*(i-
1)+1:nbeads*(i-1)+nbeads,2),r(nbeads*(i-1)+1:nbeads*(i-
1)+nbeads,3),'k-');hold on; drawnow;

   end

   end

   if nd   == 3

      axis([-2 8 -5 5 -5 5])
      az    = 8;
      el    = 15;
      view(az,el);
      xlabel('X');
      ylabel('Y');
      wx    = [0 0];
      wx2   = wx-.1;
      wy    = [0 0];
      wz    = [-5 5];
      plot3(wx,wy,wz,'b-',wx2,wy,wz,'b-');hold on;
      drawnow; % Draw Wall

   end

     hold off;
     nplot = nplot+incp;
     it

 end

end
```

Chem.m

```
% Function to Calculate Chemical Interaction
%
% States: 0:free, 1: singly bound, 2: doubly bound
%
% Brian T. Burgess, 6-13-2000

function [link,state,fxlink]     = chem(ip,nbeads,
                                   ic,vps,state,link,fxlink,A,
                                   Kf,Kr,dt,alpha,Gam2,nfmt,
                                   nd,r,rmin2)


%%% Chemical Interaction Force
fchem=0;

%%%% Test For Free Actin Bead
if state(nbeads*(ic-1)+ip,1)     == 0

%%% Check to see if Bead can X-Link with any neighboring beads
[state,fxlink,link,xlk]          = xlink(link,state,
                                   fxlink,nbeads,nfmt,r,Gam2,
                                   rmin2,ic,ip,nd,Kf,A,dt,
                                   vps,alpha);

%%% If no X-Linking occurred, Test for Singly Bound Actinin
     if xlk      ==0

          if rand(1)<(Kf*A*dt)
          %%% If Bound, Change State Label
          % Bead is Singly bound
          state(nbeads*(ic-1)+ip,1)=1;
          end

     end

     %%%% Tests For Singly Bound Actin Bead
     elseif state(nbeads*(ic-1)+ip,1) ==1

     %%% Test for X-Linking for Singly Bound
     [state,fxlink,link,xlk]          = xlink(link,state,fxlink,
                                        nbeads,nfmt,r,Gam2,
                                        rmin2,ic,ip,nd,Kf,
                                        A,dt,vps,alpha);

     %%% Test for Disassociation for Singly Bound
     if xlk           ==0             % If True, No X-Linking

     if rand(1)       <=Kr*dt
```

```
      state(nbeads*(ic-1)+ip)=0;

end

end


%%%% Test for X-Linked Beads
else
if state(nbeads*(ic-1)+ip,1)==2

      %%% Test for Disassociation
      if rand(1)                          <=Kr*dt

            % Set State to Free
            state(nbeads*(ic-1)+ip,1)   = 0;
            % Record Bead it is linked to
            clk   = link(nbeads*(ic-1)+ip,1);
            % Set Linkage to Zero
            link(nbeads*(ic-1)+ip,1)    = 0;

            fxlink(nbeads*(ic-1)+ip,vps)= zeros(1,nd);

            % Change State of other bead to Singly Bound
            state(clk,1)                = 1;
            % Set Linkage to Zero
            link(clk,1)                 = 0;

            fxlink(clk,vps)             = zeros(1,nd);

            %%% If No Disassociation, Then Calc Force
      else
            lnum=link(nbeads*(ic-1)+ip,1);
            dist=r(lnum,vps)-r(nbeads*(ic-1)+ip,vps);
            dmag=sqrt(sum(dist.^2));
            fmag=Gam2*(dmag-rmin2);
            fchem=dist./dmag.*fmag;
            if nd==2
                  fxlink(nbeads*(ic-1)+ip,vps)= [fchem(1)
                                                 fchem(2)];
                  % Force in Neg Direction for other bead
                  fxlink(lnum,vps)            = [-fchem(1)
                                                 -fchem(2)];

        elseif nd      ==3
            fxlink(nbeads*(ic-1)+ip,vps)=[fchem(1)
                                          fchem(2) fchem(3)];
            fxlink(lnum,vps)=[-fchem(1) -fchem(2) -fchem(3)];
        end

      end
```

```
        end

        clear fmag dist lnum

    end
```

xlink.m

```
% Function to Calculate if X-Linking Occurs with a Bead
% If X-Link Occurs, Chemical Force due to X-Linking is Calculated
%
%
% Brian T. Burgess
%
%

%%% Check to see if Bead can X-Link with any neighboring beads
function [state,fxlink,link,xlk] =xlink(link,
state,fxlink,nbeads,nfmt,r,Gam2,rmin2,ic,ip,nd,Kf,A,dt,vps,alpha)

% Check to see if curr bead is either free or singly bound
xlk                          = 0;
if state(nbeads*(ic-1)+ip,1)     == 0
    % Can only X-link to Singly Bound Actin Molecules
    bb                       = sparse(state==1);

else
    % Can only X-link to Free Actin Molecules
    bb                       =sparse(state==0);

end

% Calculate Sep Dist of Curr Bead from all Other Beads
ml          =ones(nbeads*nfmt,1);
for ii      = 1:nfmt

    if ic       == ii

        ml(nbeads*(ii-1)+1:nbeads*(ii
        1)+nbeads)=ml(nbeads*(ii-1)+1:nbeads*(ii-1)+nbeads)*0;

    end;

end;

atest2      = bb.*ml;
c           = find(atest2);

if nnz(c)   ~= 0
```

```
dX          = r(c,1)-r(nbeads*(ic-1)+ip,1);
dY          = r(c,2)-r(nbeads*(ic-1)+ip,2);
if nd==2
     dst    = [dX dY];
     dst2   = (dX).^2+(dY).^2;
elseif nd   == 3
     dZ     = r(c,3)-r(nbeads*(ic-1)+ip,3);
     dst    = [dX dY dZ];
     dst2   = (dX).^2+(dY).^2+(dZ).^2;

  end
% Squared Sep Dist
hid         = dst.^2;

% Magnitude of Sep Dist
dmag        = sqrt(sum(hid,2));
sz=size(dmag);

% If Sep Dist is >2*length of actinin, restrict from
% From Binding.  Do this because, w/ random # generator
% even extreme sep dist are still possible.  Physically
% this would not occur.

for i       = 1:sz(1)

     if dmag(i) >2*rmin2

          dmag(i)=1e99;

     end
end

%%% Test if X-Linking Occurs w/ Neighboring Beads
ss    =rand(1)<=10.*Kf.*A.*dt.*exp(-alpha^(-2).*(dmag-rmin2).^2);

% Test if X-Linking Occurred
if nnz(ss)>0
     % xlk       = 1: X-Link Ocurred;2: No X-Link
     xlk         = 1;
     ll          = min(abs(dmag));

     % Find Bead w/ Min Distance between current Bead
     % Assume bead closest to the Curr Bead
     % binds to the Curr Bead

     sz                    = size(dmag);

     for hh                = 1:sz(1)

          if abs(dmag(hh))==ll
```

```
            re          = hh;

        end
    end

    % Sep Dist between two newly crosslink beads
    dist=[dX(re) dY(re)];

    if nd           == 3

        dist        = [dist dZ(re)];

    end

    % Stores Bead Location Current Bead is X-Linked to
    clk             = c(re);
    dmag            = sqrt(sum(dist.^2));

    % Force Due to X-Linking
    fmag            = Gam2*(dmag-rmin2);
    fchem           = (dist./dmag).*fmag;

    if nd           == 2
    fxlink(nbeads*(ic-1)+ip,vps)    = [fchem(1) fchem(2)];
    fxlink(clk,vps)                 = [-fchem(1) -fchem(2)];
    elseif nd                       == 3
    fxlink(nbeads*(ic-1)+ip,vps)    = [fchem(1) fchem(2)
                                       fchem(3)];

    fxlink(clk,vps)                 = [-fchem(1) -fchem(2)
                                       fchem(3)];

    end
    fxlink;

    %%% Change State and Link Labels
    state(nbeads*(ic-1)+ip,1)  = 2;
    state(clk,1)               = 2;
    link(nbeads*(ic-1)+ip,1)   = clk;
    link(clk,1)                = nbeads*(ic-1)+ip;

    end

end
```

APPENDIX F
COMPUTER CODE FOR ANALYZING LISTERIA MOTILITY

Computer code for analyzing listeria cell displacements. Note These programs are designed to be used with Optimas image analysis software.

Listpla2.mac

```
/* This program is designed to playback in automated fashion the
images stored on hard drive for Listeria Injection Experiments.
There is a 500ms delay between loading each image.
Created 04/10/97      Brian Burgess  */

RunMacro ("C:/burgess/macros/bcomm1.mac");
RunMacro ("C:/burgess/macros/bscini.mac");
RunMacro ("C:/burgess/macros/bscini2.mac");

mwindow(0);
pictureroi      = roi;
field           = 1;
date            = 84;

 field          = prompt("Enter Field Number To View, Just Viewed
Field
":totext(field), "integer");
 i                    = 1;

while (field<99)
 {
     i              = 1;
     while (i<10)
     {

          openimage("d:/burgess/im804/d":totext(date):
          totext(field):"0":totext(i):".tif",pictureroi);
          delayms(500);
          i         = i+1;
     }

     delayms(1000);
     field          = prompt("Enter Field Number To View, Just
Viewed
```

```
    Field ":totext(field), "integer");
]
```

Anzelis2.mac

```
/* This program is designed to playback images stored to the hard
drive and analyze the cell positions storing the positions in
calibrated units to a Excel spreadsheet.

Created 12/15/97      Brian Burgess  */

RunMacro ("C:/burgess/macros/bcomm1.mac");
RunMacro ("C:/burgess/macros/bscini2.mac");
RunMacro ("C:/burgess/macros/bscini.mac");

calibrate(Southwick63x);
Show("Calibration Set to 63x Hammutsu Camera");

Pictureroi            = roi;

/* Initialize Excel column value */
column                = 2;
NGrid                 = 4:4;

/* Set Imaging Parameters*/
filters(sharpenhigh);

TempThres             = Threshold();
MinPoints             = Prompt("Enter min. number
                        pixels/cell","INTEGER");
REAL RMP;
RMP                   = MinPoints;
AreaCNVFactors[5]     = RMP;
AreaCNVFactors[7]     = TRUE;

Th                    = Threshold();
Th2                   = Threshold();
freeze();
field                 = 1;

/* Heading For the Spreadsheet  */
sheet           = "Sheet1";
hChan           = DDEInitiate("Excel",sheet);
DDEPoke (hChan,"R1C1:R1C5", "This Data Corresponds to Field
":totext(field):" Images");

DDETerminate(hChan);

date            = 84;
```

```
m               =roi;

clearscreen();
field       = prompt("Enter Field Number To View, Just Viewed
Field
":totext(field), "integer");

  while (field<99)
      [
      i           = 1;
      Show("Field ":totext(field));

      /* Heading For the Spreadsheet  */

      sheet       = "Sheet1";
      hChan=DDEInitiate("Excel",sheet);
      DDEPoke (hChan,"R1C1:R1C5", "This Data Corresponds to Field
      ":totext(field):" Images");
      DDETerminate(hChan);
      imagecount = 0;

      while (i<10)
            [
            openimage("d:/burgess/im804/d":totext(date):totext(fie
            ld):"0
            :totext(i):".tif",pictureroi);
            delayms(500);
            i                 = i+1;

            /* increment to the next picture, same field. */
            imagecout         = 1;
            imageRoi          = SelectRoi();

            filters(sharpenhigh);
            // filters(average5x5);
            threshold(th);

            SetExport(PtPoints,1,TRUE); /* set "To DDE" */
            SetExport(Arcentroid,1,TRUE);
            Roi               = imageroi;
            AutoExtract       = True;
            CreateArea(,FALSE , TRUE);
            delayms(1000);

            itest             = (GetShape(MArCentroid));

             if (itest[0]==0)
                  {
                  MacroMessage("Did Not Determine Area Centroid for
Cell.
                  You will Need to Locate Manually the Center of
            Mass");
```

```
                CreatePoints();
                Ans         = Prompt("Is this Point
Satisfactory?",2);

                While (ans==FALSE)
                      {
                      clearscreen();
                      CreatePoints();
                      Ans   = Prompt("Is this Point
Satisfactory?",2);
                      }

                coordinates=ptpoints;
                clearscreen();
                }
                else
                {
                 CreatePoints(mArCentroid);
                 Coordinates     = marcentroid;
                }

/*  Setup Excel Program  */
Real m;
m     = getshape(coordinates);
/* k is the row counter for data within one field */
k     = 1;
/* x and y are extracting the x and y positions from H vector */
x     = 0;
y     = 1;
EFile = "Sheet1";
While (k<(m+1))
      {
      Exl   ="R":totext(2+k):"C":totext(column):":
             R":totext(2+k):"C":totext(column+2);
      hChan=DDEInitiate("Excel",EFile);
      DDEPoke (hChan,Exl,coordinates[x:y]);
      DDETerminate(hChan);
      /* increment x and y to the next "pair" of x and y values
*/
      x           = x+2;

      y           = y+2;
      /* move to next row down */
      k           = k+1;

      }

       /* Place New Data Point in Next column */
      column      = column + 3;

      clearscreen();
```

```
     delayms(500);
}
/* Increment to Next Data Set */
Show("Just Viewed Field ":totext(field));
Field           =  Prompt(" Just Viewed Field ":totext(field): "
Enter
                     Field number to do Next", "INTEGER");

Show("Now moving to Field ":totext(field));

column          = 2;

/* Reset to zero enabling new roi for image processing */
imagecount      = 0;

}
```

APPENDIX G
DERIVATION OF LOCAL BENDING ENERGY, $F_{B,I}$

Mechanical energy due to bending an actin filament can be modeled described by:

$$E_{\mathrm{B}} = \frac{1}{2}\int_0^L B\frac{d^2\mathbf{r}}{ds^2} \qquad \text{A1}$$

where d**r** is given by:

$$d\mathbf{r} = \frac{\mathbf{r}_i - \mathbf{r}_{i-1}}{\Delta r_{\min}} \qquad \text{A2}$$

where $\Delta r_{min} \approx ds$ and **r** is a the vector position of the i^{th} bead along the filament. $d\mathbf{r}/ds$ is then:

$$\frac{d\mathbf{r}}{ds} = \frac{1}{\Delta r_{\min}^2}\left[\mathbf{r}_{i-1} - 2\mathbf{r}_i + \mathbf{r}_{i+1}\right] \qquad \text{A3}$$

Substituting A3 into A1 gives:

$$E_B = \frac{B}{2\Delta r_{\min}^3}\sum_{J=1}^{N}(r_{i-1} - 2r_i + r_{i+1})^2 \qquad \text{A4}$$

The local bending force on the i^{th} bead can then be calculated by:

$$F_{B,i} = -\frac{dE_{B,i}}{d\mathbf{r}_i} \qquad \text{A5}$$

Differentiating A5 with respect to r_i:

$$F_{B,i} = \frac{B}{2\Delta r_{\min}^3}\frac{d}{dr_i}\left(\left[\sum_{J=1}^{N} r_{i-1} - 2r_i + r_{i+1}\right]^2\right) \qquad \text{A6}$$

Solving A6 gives:

$i = 1$:

$$F_{B.1} = -\frac{B}{r_{\min}^3}\left[\mathbf{r}_1 - 2\mathbf{r}_2 + \mathbf{r}_3\right] \qquad \text{A7}$$

$i = 2$:

$$F_{B.2} = -\frac{B}{r_{\min}^3}\left[-2\mathbf{r}_1 + 5\mathbf{r}_2 - 4\mathbf{r}_3 + \mathbf{r}_4\right] \qquad \text{A8}$$

$2 < i < N\text{-}1$:

$$F_{B.i} = -\frac{B}{r_{\min}^3}\left[\mathbf{r}_{i-2} - 4\mathbf{r}_{i-1} + 6\mathbf{r}_i + \mathbf{r}_{i+1}\right] \qquad \text{A9}$$

$i = N\text{-}1$:

$$F_{B.N-1} = -\frac{B}{r_{\min}^3}\left[\mathbf{r}_{N-3} - 4\mathbf{r}_{N-2} + 5\mathbf{r}_{N-1} - 2\mathbf{r}_N\right] \qquad \text{A10}$$

$i = N$:

$$F_{B.N} = -\frac{B}{r_{\min}^3}\left[\mathbf{r}_{N-2} - 2\mathbf{r}_{N-1} + \mathbf{r}_N\right] \qquad \text{A11}$$

APPENDIX H
BUNDLING VELOCITY OF ALPHA-ACTININ

The free bundling velocity, V_B, of α-actinin can be described:

$$V_B = k_{f,2} U_1 (1 - U_1) \qquad \text{A12}$$

where k_{f2} is bundling velocity (nm/s), U_1 probability associated with form a singly-bound actinin molecule and $1\text{-}U_1$ is probability the α-actinin molecule not disassociating. The bundling velocity, k_{f2} , can be described by:

$$k_{f,2} = \frac{k_{f,1} \exp(-\frac{(z - z_0)^2}{2\sigma^2})}{(2\pi)^{3/2} z\sigma} \qquad \text{A13}$$

where k_{f1} $(Ms)^{-1}$ reaction binding affinity of an α-actinin molecule binding to an actin filament described by Eqn. 4, z is the separation distance between two filaments, z_0 is the optimal binding separation distance, σ is the standard deviation of the length of α-actinin molecules due to Brownian motion (modeled as harmonic spring as described in text). Assuming a free cytoplasmic α-actinin concentration of ≈3e-7M and binding affinities as $k_r \approx .4s^{-1}$, $k_f \approx 6.8e5$ $(Ms)^{-1}$ (Wachsstock et al., 1993; Goldmann and Isenberg, 1993), $\sigma \approx 25$nm, and $z = z_0$ (optimal binding separation distance), the free bundling velocity, $V_B \approx 7$nm/s.

REFERENCES

Akiyama, S. K. 1996. Integrins in cell adhesion and signaling. *Hum Cell.* 9:181-6.

Albelda, S. M., S. A. Mette, D. E. Elder, R. Stewart, L. Damjanovich, M. Herlyn, and C. A. Buck. 1990. Integrin distribution in malignant melanoma: association of the beta 3 subunit with tumor progression. *Cancer Res.* 50:6757-64.

Alt, W. 1990. Correlation Analysis of Two-Dimensional Locomotion Paths. *Biological Motion. W. Alt and G. Hoffmann, Berlin, Springer-Verlag.* 89:254-268.

Ayscough, K. R. 1998. In vivo functions of actin-binding proteins. *Curr Opin Cell Biol.* 10:102-11.

Aznavoorian, S., M. L. Stracke, H. Krutzsch, E. Schiffmann, and L. A. Liotta. 1990. Signal transduction for chemotaxis and haptotaxis by matrix molecules in tumor cells. *J Cell Biol.* 110:1427-38.

Aznavoorian, S., M. L. Stracke, J. Parsons, J. McClanahan, and L. A. Liotta. 1996. Integrin alphavbeta3 mediates chemotactic and haptotactic motility in human melanoma cells through different signaling pathways. *J Biol Chem.* 271:3247-54.

Baron, M. D., M. D. Davison, P. Jones, and D. R. Critchley. 1987. The sequence of chick alpha-actinin reveals homologies to spectrin and calmodulin. *J Biol Chem.* 262:17623-9.

Bartles, J. R. 2000. Parallel actin bundles and their multiple actin-bundling proteins. *Curr Opin Cell Biol.* 12:72-8.

Bausch, A. R., W. Moller, and E. Sackmann. 1999. Measurement of local viscoelasticity and forces in living cells by magnetic tweezers. *Biophys J.* 76:573-9.

Bergman, A., and K. Zygourakis . 1998. Lymphocyte Motility and Adhesion on Fibronectin Surfaces Through the Beta-1 Integrin. *1998 AICHE Annual Meeting.* Paper 278a.

Berman, A., G. Morozevich, I. Karmansky, A. Gleiberman, and V. Bychkova. 1993. Adhesion of mouse hepatocytes to type I collagen: role of supramolecular forms and effect of proteolytic degradation. *Biochem Biophys Res Commun.* 194:351-7.

Borisy, G. G., and T. M. Svitkina. 2000. Actin machinery: pushing the envelope. *Curr Opin Cell Biol.* 12:104-12.

Brandley, B. K., and R. L. Schnaar. 1989. Tumor cell haptotaxis on covalently immobilized linear and exponential gradients of a cell adhesion peptide. *Dev Biol.* 135:74-86.

Bremer, A., and U. Aebi. 1992. The structure of the F-actin filament and the actin molecule. *Curr Opin Cell Biol.* 4:20-6.

Bretscher, A., J. Vandekerckhove, and K. Weber. 1979. alpha-Actinins from chicken skeletal muscle and smooth muscle show considerable chemical and immunological differences. *Eur J Biochem.* 100:237-43.

Buettner, H. M., R. N. Pittman, and J. K. Ivins. 1994. A model of neurite extension across regions of nonpermissive substrate: simulations based on experimental measurement of growth cone motility and filopodial dynamics. *Dev Biol.* 163:407-22.

Bultmann, B. D., and H. Gruler. 1983. Analysis of the directed and nondirected movement of human granulocytes: influence of temperature and ECHO 9 virus on N-formylmethionylleucylphenylalanine-induced chemokinesis and chemotaxis. *J Cell Biol.* 96:1708-16.

Burgess, B. T., J. M. Myles, and R. B. Dickinson. 1999. A mechanistic model relating cell migration to substratum adhesiveness in a three dimensional matrix. *In* American Institute of Chemical Engineers, Dallas, Texas.

Burgess, B. T., J. M. Myles, and R. B. Dickinson. 2000. Quantitative analysis of adhesion-mediated cell migration in three-dimensional gels of RGD-grafted collagen. *Annals of Biomed Eng.* 28:1-9.

Carter, S. B. 1965. Principles of cell motility: the direction of cell movement and cancer invasion. *Nature.* 208:1183-7.

Condeelis, J. 1993. Life at the leading edge: the formation of cell protrusions. *Annu Rev Cell Biol.* 9:411-44.

Cudmore, S., P. Cossart, G. Griffiths, and M. Way. 1995. Actin-based motility of vaccinia virus. *Nature.* 378:636-8.

Dabiri, G. A., J. M. Sanger, D. A. Portnoy, and F. S. Southwick. 1990. Listeria monocytogenes moves rapidly through the host-cell cytoplasm by inducing directional actin assembly. *Proc Natl Acad Sci U S A.* 87:6068-72.

Darnell, J., H. Lodish, and D. Baltimore. 1990. Molecular Cell Biology. W.H. Freeman and Company, New York.

Dedhar, S., E. Ruoslahti, and M. D. Pierschbacher. 1987. A cell surface receptor complex for collagen type I recognizes the Arg- Gly-Asp sequence. *J Cell Biol.* 104:585-93.

Deryugina, E. I., M. A. Bourdon, R. A. Reisfeld, and A. Strongin. 1998. Remodeling of collagen matrix by human tumor cells requires activation and cell surface association of matrix metalloproteinase-2. *Cancer Res.* 58:3743-50.

Dickinson, R. B., S. Guido, and R. T. Tranquillo. 1994. Biased cell migration of fibroblasts exhibiting contact guidance in oriented collagen gels. *Ann Biomed Eng.* 22:342-56.

Dickinson, R. B., and R. T. Tranquillo. 1993a. Optimal estimation of cell movement indices from the statistical analysis of cell tracking data. *AICHE J.* 39:1995-2010.

Dickinson, R. B., and R. T. Tranquillo. 1993b. A stochastic model for adhesion-mediated cell random motility and haptotaxis. *J Math Biol.* 31:563-600.

DiMilla, P. A., K. Barbee, and D. A. Lauffenburger. 1991. Mathematical model for the effects of adhesion and mechanics on cell migration speed [published erratum appears in Biophys J 1991 Oct;60(4):983]. *Biophys J.* 60:15-37.

DiMilla, P. A., J. A. Stone, J. A. Quinn, S. M. Albelda, and D. A. Lauffenburger. 1993. Maximal migration of human smooth muscle cells on fibronectin and type IV collagen occurs at an intermediate attachment strength. *J Cell Biol.* 122:729-37.

Doi, M., and S. F. Edwards. 1986. The Theory of Polymer Dynamics. Clarendon Press, Oxford.

Dold, F. G., J. M. Sanger, and J. W. Sanger. 1994. Intact alpha-actinin molecules are needed for both the assembly of actin into the tails and the locomotion of Listeria monocytogenes inside infected cells [see comments]. *Cell Motil Cytoskeleton.* 28:97-107.

Duband, J. L., S. Dufour, S. S. Yamada, K. M. Yamada, and J. P. Thiery. 1991. Neural crest cell locomotion induced by antibodies to beta 1 integrins. A tool for studying the roles of substratum molecular avidity and density in migration. *J Cell Sci.* 98:517-32.

Dunn, G. A. 1983. Characterising a kinesis response: time averaged measures of cell speed and directional persistence. *Agents Actions Suppl.* 12:14-33.

Erickson, C. A. 1985. Control of neural crest cell dispersion in the trunk of the avian embryo. *Dev Biol.* 111:138-57.

Federal Drug Agency (FDA). 1999. Internet: http://www.fda.cfsan.fdagov.

Friederich, E., C. Huet, M. Arpin, and D. Louvard. 1989. Villin induces microvilli growth and actin redistribution in transfected fibroblasts. *Cell.* 59:461-75.

Friedl, P., E. B. Brocker, and K. S. Zanker. 1998a. Integrins, cell matrix interactions and cell migration strategies: fundamental differences in leukocytes and tumor cells. *Cell Adhes Commun.* 6:225-36.

Friedl, P., F. Entschladen, C. Conrad, B. Niggemann, and K. S. Zanker. 1998b. CD4+ T lymphocytes migrating in three-dimensional collagen lattices lack focal adhesions and utilize beta1 integrin-independent strategies for polarization, interaction with collagen fibers and locomotion. *Eur J Immunol.* 28:2331-43.

Friedl, P., K. Maaser, C. E. Klein, B. Niggemann, G. Krohne, and K. S. Zanker. 1997. Migration of highly aggressive MV3 melanoma cells in 3-dimensional collagen

lattices results in local matrix reorganization and shedding of alpha2 and beta1 integrins and CD44. *Cancer Res.* 57:2061-70.

Friedl, P., K. S. Zanker, and E. B. Brocker. 1998c. Cell migration strategies in 3-D extracellular matrix: differences in morphology, cell matrix interactions, and integrin function [In Process Citation]. *Microsc Res Tech.* 43:369-78.

Gailit, J., C. Clarke, D. Newman, M. G. Tonnesen, M. W. Mosesson, and R. A. Clark. 1997. Human fibroblasts bind directly to fibrinogen at RGD sites through integrin alpha(v)beta3. *Exp Cell Res.* 232:118-26.

Goldmann, W. H., and G. Isenberg. 1993. Analysis of filamin and alpha-actinin binding to actin by the stopped flow method. *FEBS Lett.* 336:408-10.

Goodman, S. L., M. Aumailley, and H. von der Mark. 1991. Multiple cell surface receptors for the short arms of laminin: alpha 1 beta 1 integrin and RGD-dependent proteins mediate cell attachment only to domains III in murine tumor laminin. *J Cell Biol.* 113:931-41.

Götter, R., K. Kroy, E. M. Frey, M. Bärmann, and S. E. 1996. Dynamic Light Scattering from Semidilute Actin Solutions: A Study of Hydrodynamic Screening, Filament Bending Stiffness, and the Effect of Tropomyosin/Troponin-Binding. *Macromolecules.* 29:30-36.

Gouin, E., H. Gantelet, C. Egile, I. Lasa, H. Ohayon, V. Villiers, P. Gounon, P. J. Sansonetti, and P. Cossart. 1999. A comparative study of the actin-based motilities of the pathogenic bacteria Listeria monocytogenes, Shigella flexneri and Rickettsia conorii. *J Cell Sci.* 112:1697-708.

Grazi, E., P. Cuneo, E. Magri, and C. Schwienbacher. 1992. Preferential binding of alpha-actinin to actin bundles. *FEBS Lett.* 314:348-50.

Grinnell, F. 1982. Migration of human neutrophils in hydrated collagen lattices. *J Cell Sci.* 58:95-108.

Guido, S., and R. T. Tranquillo. 1993. A methodology for the systematic and quantitative study of cell contact guidance in oriented collagen gels. Correlation of fibroblast orientation and gel birefringence. *J Cell Sci.* 105:317-31.

Gunzer, M., E. Kampgen, E. B. Brocker, K. S. Zanker, and P. Friedl. 1997. Migration of dendritic cells in 3D-collagen lattices. Visualisation of dynamic interactions with the substratum and the distribution of surface structures via a novel confocal reflection imaging technique. *Adv Exp Med Biol.* 417:97-103.

Haas, T. L., S. J. Davis, and J. A. Madri. 1998. Three-dimensional type I collagen lattices induce coordinate expression of matrix metalloproteinases MT1-MMP and MMP-2 in microvascular endothelial cells. *J Biol Chem.* 273:3604-10.

Harris, A. 1973. Behavior of cultured cells on substrata of variable adhesiveness. *Exp Cell Res.* 77:285-97.

Hauzenberger, D., J. Klominek, and K. G. Sundqvist. 1994. Functional specialization of fibronectin-binding beta 1-integrins in T lymphocyte migration. *J Immunol.* 153:960-71.

Huttenlocher, A., M. H. Ginsberg, and A. F. Horwitz. 1996. Modulation of cell migration by integrin-mediated cytoskeletal linkages and ligand-binding affinity. *J Cell Biol.* 134:1551-62.

Huttenlocher, A., R. R. Sandborg, and A. F. Horwitz. 1995. Adhesion in cell migration. *Curr Opin Cell Biol.* 7:697-706.

Imamura, M., T. Endo, M. Kuroda, T. Tanaka, and T. Masaki. 1988. Substructure and higher structure of chicken smooth muscle alpha- actinin molecule. *J Biol Chem.* 263:7800-5.

Isambert, H., P. Venier, A. C. Maggs, A. Fattoum, R. Kassab, D. Pantaloni, and M. F. Carlier. 1995. Flexibility of actin filaments derived from thermal fluctuations. Effect of bound nucleotide, phalloidin, and muscle regulatory proteins. *J Biol Chem.* 270:11437-44.

Kas, J., H. Strey, J. X. Tang, D. Finger, R. Ezzell, E. Sackmann, and P. A. Janmey. 1996. F-actin, a model polymer for semiflexible chains in dilute, semidilute, and liquid crystalline solutions. *Biophys J.* 70:609-25.

Keely, P. J., A. M. Fong, M. M. Zutter, and S. A. Santoro. 1995. Alteration of collagen-dependent adhesion, motility, and morphogenesis by the expression of antisense alpha 2 integrin mRNA in mammary cells. *J Cell Sci.* 108:595-607.

Keller, H. U., J. H. Wissler, and J. Ploem. 1979. Chemotaxis is not a special case of haptotaxis. *Experientia.* 35:1669-71.

Klominek, J., S. Sumitran Karuppan, and D. Hauzenberger. 1997. Differential motile response of human malignant mesothelioma cells to fibronectin, laminin and collagen type IV: the role of beta1 integrins. *Int J Cancer.* 72:1034-44.

Kramer, R. H., and N. Marks. 1989. Identification of integrin collagen receptors on human melanoma cells. *J Biol Chem.* 264:4684-8.

Kramer, R. H., M. Vu, Y. F. Cheng, and D. M. Ramos. 1991. Integrin expression in malignant melanoma. *Cancer Metastasis Rev.* 10:49-59.

Kuntz, R. M., and W. M. Saltzman. 1997. Neutrophil motility in extracellular matrix gels: mesh size and adhesion affect speed of migration. *Biophys J.* 72:1472-80.

Lauffenburger, D. 1983. Measurement of phenomenological parameters for leukocyte motility and chemotaxis. *Agents Actions Suppl.* 12:34-53.

Lauffenburger, D. A., and A. F. Horwitz. 1996. Cell migration: a physically integrated molecular process. *Cell.* 84:359-69.

Lin, T. H., A. E. Aplin, Y. Shen, Q. Chen, M. Schaller, L. Romer, I. Aukhil, and R. L. Juliano. 1997. Integrin-mediated activation of MAP kinase is independent of FAK: evidence for dual integrin signaling pathways in fibroblasts. *J Cell Biol.* 136:1385-95.

Loisel, T. P., R. Boujemaa, D. Pantaloni, and M. F. Carlier. 1999. Reconstitution of actin-based motility of Listeria and Shigella using pure proteins [see comments]. *Nature.* 401:613-6.

McCarthy, J. B., and L. T. Furcht. 1984. Laminin and fibronectin promote the haptotactic migration of B16 mouse melanoma cells in vitro. *J Cell Biol.* 98:1474-80.

McCarthy, J. B., S. L. Palm, and L. T. Furcht. 1983. Migration by haptotaxis of a Schwann cell tumor line to the basement membrane glycoprotein laminin. *J Cell Biol.* 97:772-7.

McNeil, P. L., and E. Warder. 1987. Glass beads load macromolecules into living cells. *J Cell Sci.* 88:669-78.

Meyer, R. K., and U. Aebi. 1990. Bundling of actin filaments by alpha-actinin depends on its molecular length. *J Cell Biol.* 110:2013-24.

Mimura, N., and A. Asano. 1986. Isolation and characterization of a conserved actin-binding domain from rat hepatic actinogelin, rat skeletal muscle, and chicken gizzard alpha-actinins. *J Biol Chem.* 261:10680-7.

Mogilner, A., and G. Oster. 1996. Cell motility driven by actin polymerization. *Biophys J.* 71:3030-45.

Morla, A., Z. Zhang, and E. Ruoslahti. 1994. Superfibronectin is a functionally distinct form of fibronectin. *Nature.* 367:193-6.

Myles, J. M., B. T. Burgess, and R. B. Dickinson. 2000. Modification of the adhesive properties of collagen by covalent grafting with RGD peptides. *J. Biomater Sci. Polymer Edn.* 11:69-86.

Nanavati, D., F. T. Ashton, J. M. Sanger, and J. W. Sanger. 1994. Dynamics of actin and alpha-actinin in the tails of Listeria monocytogenes in infected PtK2 cells [see comments]. *Cell Motil Cytoskeleton.* 28:346-58.

Nuccitelli, R., and C. A. Erickson. 1983. Embryonic cell motility can be guided by physiological electric fields. *Exp Cell Res.* 147:195-201.

Oliver, T., M. Dembo, and K. Jacobson. 1999. Separation of propulsive and adhesive traction stresses in locomoting keratocytes. *J Cell Biol.* 145:589-604.

O'Toole, T. E., J. C. Loftus, X. P. Du, A. A. Glass, Z. M. Ruggeri, S. J. Shattil, E. F. Plow, and M. H. Ginsberg. 1990. Affinity modulation of the alpha IIb beta 3 integrin (platelet GPIIb- IIIa) is an intrinsic property of the receptor. *Cell Regul.* 1:883-93.

Ottinger, H. C. 1996. Stochastic Processes in Polymeric Fluids : Tools and Examples for Developing Simulation Algorithms. Springer-Verlag, Berlin/Heidelberg.

Palecek, S. P., A. Huttenlocher, A. F. Horwitz, and D. A. Lauffenburger. 1998. Physical and biochemical regulation of integrin release during rear detachment of migrating cells. *J Cell Sci.* 111:929-40.

Palecek, S. P., J. C. Loftus, M. H. Ginsberg, D. A. Lauffenburger, and A. F. Horwitz. 1997. Integrin-ligand binding properties govern cell migration speed through cell-substratum adhesiveness [published erratum appears in Nature 1997 Jul 10;388(6638):210]. *Nature.* 385:537-40.

Pavalko, F. M., and K. Burridge. 1991. Disruption of the actin cytoskeleton after microinjection of proteolytic fragments of alpha-actinin. *J Cell Biol.* 114:481-91.

Peskin, C. S., G. M. Odell, and G. F. Oster. 1993. Cellular motions and thermal fluctuations: the Brownian ratchet. *Biophys J.* 65:316-24.

Pfaff, M., M. Aumailley, U. Specks, J. Knolle, H. G. Zerwes, and R. Timpl. 1993. Integrin and Arg-Gly-Asp dependence of cell adhesion to the native and unfolded triple helix of collagen type VI. *Exp Cell Res.* 206:167-76.

Pierschbacher, M. D., E. G. Hayman, and E. Ruoslahti. 1985. The cell attachment determinant in fibronectin. *J Cell Biochem.* 28:115-26.

Pierschbacher, M. D., and E. Ruoslahti. 1984. Variants of the cell recognition site of fibronectin that retain attachment-promoting activity. *Proc Natl Acad Sci U S A.* 81:5985-8.

Podlubnaya, Z. A., L. A. Tskhovrebova, M. M. Zaalishtsbvili, and G. A. Stefanenko. 1975. Electron microscopic study of alpha-actinin. *J Mol Biol.* 92:357-9.

Pollard, T. D. 1990. Actin. *Curr Opin Cell Biol.* 2:33-40.

Pollard, T. D., and R. D. Goldman. 1993. Cytoplasm and Cell Motility. *Curr Opin Cell Biol.* 5:1-2.

Pytela, R., M. D. Pierschbacher, M. H. Ginsberg, E. F. Plow, and E. Ruoslahti. 1986. Platelet membrane glycoprotein IIb/IIIa: member of a family of Arg-Gly- Asp--specific adhesion receptors. *Science.* 231:1559-62.

Rich, A., and A. K. Harris. 1981. Anomalous preferences of cultured macrophages for hydrophobic and roughened substrata. *J Cell Sci.* 50:1-7.

Roberts, T. M., and M. Stewart. 2000. Acting like actin. The dynamics of the nematode major sperm protein (msp) cytoskeleton indicate a push-pull mechanism for amoeboid cell motility. *J Cell Biol.* 149:7-12.

Ruoslahti, E. 1991. Integrins. *J Clin Invest.* 87:1-5.

Ruoslahti, E. 1996. RGD and other recognition sequences for integrins. *Annu Rev Cell Dev Biol.* 12:697-715.

Ruoslahti, E. 1997. Integrins as signaling molecules and targets for tumor therapy. *Kidney Int.* 51:1413-7.

Ruoslahti, E., and M. D. Pierschbacher. 1986. Arg-Gly-Asp: a versatile cell recognition signal. *Cell.* 44:517-8.

Ruoslahti, E., and M. D. Pierschbacher. 1987. New perspectives in cell adhesion: RGD and integrins. *Science.* 238:491-7.

Ruoslahti, E., S. Suzuki, E. G. Hayman, C. R. Ill, and M. D. Pierschbacher. 1987. Purification and characterization of vitronectin. *Methods Enzymol.* 144:430-7.

Sanger, J. M., J. W. Sanger, and F. S. Southwick. 1992. Host cell actin assembly is necessary and likely to provide the propulsive force for intracellular movement of Listeria monocytogenes. *Infect Immun.* 60:3609-19.

Schnitzer, M. J., and R. L. Juliano. 1990. Strategies for chemotaxis. In: Armitage, J.P., Lackie, J.M. Biology of the chemotactic response. :15-34.

Sechi, A. S., J. Wehland, and J. V. Small. 1997. The isolated comet tail pseudopodium of Listeria monocytogenes: a tail of two actin filament populations, long and axial and short and random. *J Cell Biol.* 137:155-67.

Sheetz, M. P. 1994. Cell migration by graded attachment to substrates and contraction. *Semin Cell Biol.* 5:149-55.

Sims, P. J., M. H. Ginsberg, E. F. Plow, and S. J. Shattil. 1991. Effect of platelet activation on the conformation of the plasma membrane glycoprotein IIb-IIIa complex. *J Biol Chem.* 266:7345-52.

Small, J. V., A. Rohlfs, and M. Herzog. 1993. Actin and cell movement. *Symp Soc Exp Biol.* 47:57-71.

Small, J. V., K. Rottner, and I. Kaverina. 1999. Functional design in the actin cytoskeleton. *Curr Opin Cell Biol.* 11:54-60.

Southwick, F. S., and D. L. Purich. 1994. Arrest of Listeria movement in host cells by a bacterial ActA analogue: implications for actin-based motility. *Proc Natl Acad Sci U S A.* 91:5168-72.

Southwick, F. S., and D. L. Purich. 1996. Mechanisms of Disease. *New Engl J of Med.* 334:770-6.

Stossel, T. P. 1993. On the crawling of animal cells. *Science.* 260:1086-94.

Suzuki, A., D. E. Goll, I. Singh, R. E. Allen, R. M. Robson, and M. H. Stromer. 1976. Some properties of purified skeletal muscle alpha-actinin. *J Biol Chem.* 251:6860-70.

Suzuki, A., D. E. Goll, M. H. Stromer, and J. Temple. 1973. -actinin from red and white porcine muscle. *Biochim Biophys Acta.* 295:188-207.

Taunton, J., B. A. Rowning, M. L. Coughlin, M. Wu, R. T. Moon, T. J. Mitchison, and C. A. Larabell. 2000. Actin-dependent propulsion of endosomes and lysosomes by recruitment of N-WASP. *J Cell Biol.* 148:519-30.

Temm-Grove, C. J., B. M. Jockusch, M. Rohde, K. Niebuhr, T. Chakraborty, and J. Wehland. 1994. Exploitation of microfilament proteins by Listeria monocytogenes: microvillus-like composition of the comet tails and vectorial spreading in polarized epithelial sheets. *J Cell Sci.* 107:2951-60.

Theriot, J. A., T. J. Mitchison, L. G. Tilney, and D. A. Portnoy. 1992. The rate of actin-based motility of intracellular Listeria monocytogenes equals the rate of actin polymerization. *Nature.* 357:257-60.

Theriot, J. A., J. Rosenblatt, D. A. Portnoy, P. J. Goldschmidt-Clermont, and T. J. Mitchison. 1994. Involvement of profilin in the actin-based motility of L. monocytogenes in cells and in cell-free extracts. *Cell.* 76:505-17.

Tilney, L. G., D. J. DeRosier, and M. S. Tilney. 1992a. How Listeria exploits host cell actin to form its own cytoskeleton. I. Formation of a tail and how that tail might be involved in movement. *J Cell Biol.* 118:71-81.

Tilney, L. G., D. J. DeRosier, A. Weber, and M. S. Tilney. 1992b. How Listeria exploits host cell actin to form its own cytoskeleton. II. Nucleation, actin filament polarity, filament assembly, and evidence for a pointed end capper. *J Cell Biol.* 118:83-93.

Tilney, L. G., and D. A. Portnoy. 1989. Actin filaments and the growth, movement, and spread of the intracellular bacterial parasite, Listeria monocytogenes. *J Cell Biol.* 109:1597-608.

Tranquillo, R. T. 1990. Theories and Models of Gradient Perception. *Biol Chemotactic Resp.* 46:35-75.

Tranquillo, R. T., and W. Alt. 1990. Glossary of terms concerning oriented movement. *Biological Motion. W. Alt and G. Hoffmann, Berlin, Springer-Verlag.* 89:510-517.

Tranquillo, R. T., T. S. Girton, B. A. Bromberek, T. G. Triebes, and D. L. Mooradian. 1996. Magnetically orientated tissue-equivalent tubes: application to a circumferentially orientated media-equivalent. *Biomaterials.* 17:349-57.

Tranquillo, R. T., S. H. Zigmond, and D. A. Lauffenburger. 1988. Measurement of the chemotaxis coefficient for human neutrophils in the under-agarose migration assay. *Cell Motil Cytoskeleton.* 11:1-15.

Valberg, P. A., and D. F. Albertini. 1985. Cytoplasmic motions, rheology, and structure probed by a novel magnetic particle method. *J Cell Biol.* 101:130-40.

Vandekerckhove, J. 1990. Actin-binding proteins. *Curr Opin Cell Biol.* 2:41-50.

Vandekerckhove, J., and K. Vancompernolle. 1992. Structural relationships of actin-binding proteins. *Curr Opin Cell Biol.* 4:36-42.

Wachsstock, D. H., W. H. Schwartz, and T. D. Pollard. 1993. Affinity of alpha-actinin for actin determines the structure and mechanical properties of actin filament gels. *Biophys J.* 65:205-14.

Wachsstock, D. H., W. H. Schwarz, and T. D. Pollard. 1994. Cross-linker dynamics determine the mechanical properties of actin gels. *Biophys J.* 66:801-9.

Welch, M. D., A. Mallavarapu, J. Rosenblatt, and T. J. Mitchison. 1997. Actin dynamics in vivo. *Curr Opin Cell Biol.* 9:54-61.

Wilkinson, P. C., J. M. Lackie, J. V. Forrester, and G. A. Dunn. 1984. Chemokinetic accumulation of human neutrophils on immune complex-coated substrata: analysis at a boundary. *J Cell Biol.* 99:1761-8.

Wong, A. J., T. D. Pollard, and I. M. Herman. 1983. Actin filament stress fibers in vascular endothelial cells in vivo. *Science.* 219:867-9.

Zaner, K. S. 1995. Physics of actin networks. I. Rheology of semi-dilute F-actin. *Biophys J.* 68:1019-26.

Zaner, K. S., and P. A. Valberg. 1989. Viscoelasticity of F-actin measured with magnetic microparticles. *J Cell Biol.* 109:2233-43.

Zeile, W. L., D. L. Purich, and F. S. Southwick. 1996. Recognition of two classes of oligoproline sequences in profilin- mediated acceleration of actin-based Shigella motility. *J Cell Biol.* 133:49-59.

Zigmond, S. H. 1981. Consequences of chemotactic peptide receptor modulation for leukocyte orientation. *J Cell Biol.* 88:644-7.

BIOGRAPHICAL SKETCH

Brian Toy Burgess was born on April 29, 1971, in Anderson, South Carolina. He is the son of Toy Gerald and Dianne Burgess and has a younger sister and brother, Ali and Kevin. He later moved with his family to Trion, Georgia; Alpharetta, Georgia; Sylacauga, Alabama and Dickson, Tennessee. In May 1990, Brian graduated with highest honors from Sylacauga High School. Thereafter he attended Auburn University from September 1990 to June 1995 where he received a bachelor's degree in Chemical Engineering. While at Auburn University he met Melanie Flynne Pilgreen, his wife to-be. Following his graduation at Auburn, Brian began graduate school in the Chemical Engineering Department at the University of Florida in August 1995. In July of 1999, Brian married Melanie Pilgreen. While working on his doctorate degree, Brian received a Master of Science degree from biomedical engineering in December 1999. Upon graduation, Brian will begin work with Merck & Co. as a senior engineer in Albany, Georgia.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Richard B. Dickinson, Chairman
Professor of Chemical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Daniel L. Purich
Professor of Biochemistry &
Molecular Biology

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Spyros Svoronos
Professor of Chemical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

Roger Tran-Son-Tay
Professor of Aerospace Engineering,
Mechanics & Engineering Science

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 2000

M. J. Ohanian
Dean, College of Engineering

Winfred M. Phillips
Dean, Graduate School